Web Information System for a Map of Leeds Pubs

Jianping ZHANG
Accounting and Computing (Industry)
Session (2006/2007)

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) _____

Summary

The aim of the project is to produce a web based database information system of a map of Leeds pubs based on Leeds CAMRA's requirements. The key objective of this project is to produce a system which will provide information for a list of CAMRA specified ale houses in Leeds in a rich interface map as it was initially proposed by a Leeds CAMRA user. A content management system will be produced to enhance the system's functionality so that Leeds CAMRA can easily maintain the system records through a friendly user interface rather than through coding.

The report will perform in-depth background research on the existing techniques and solutions, and the most suitable techniques and solutions will be chosen. Then the system will be developed with the help of the Waterfall and Prototyping models. Moreover, many professional project management techniques will be used during the development process, such as Gantt chart.

The main focuses on the system development are the functionality and the usability development. All the system functions are based on the user requirements that are gathered through a number of meeting and email contacts with the problem owner. The Nielsen's point of usability and Larisa, Thomason's website usability checklist are going to used to ensure the usability of the system.

Finally, the self-evaluation, which reflects on the project development experience with examples, will be presented in Appendix A. All the other support materials, such as SQL statements, programming code and user manual will be produced in the Appendix part.

Acknowledgments

I would like to thank the following peoples for their help during the development:

1. My supervisor Tony Jenkins and my assessor Sarah Fores for their time and guideline through the development of the project. I remember I used to doubt that the current system is too simple so that I propose a new complex system. You two spent time discussing the potential problems of my new proposal system. I insisted on my opinion for a long time. Finally I understand why my new propose system have great risk of failure with your help. Thank you for the patience because I know how annoying I can be.
2. Leeds CAMRA for providing me the chance to develop this project. I learnt lots of practical knowledge and project management skills from this system development.
3. My parents Chengben Zhang, Xiuying Chen and my girlfriend Jinling Liu for listening to my complaints and giving me support.
4. My colleague from Central Science Laboratory Jason Haritou for checking through my writing style and grammar mistakes.

Contents

Chapter 1 – Introduction…….1

1.1 Project Aim……...1

1.2 Project Objectives……1

1.3 Deliverables……1

1.4 Schedules……1

1.5 Project Overview…….3

Chapter 2 – Background research…….4

2.1 Background to the problem……4

2.2 User Requirements…….....4

2.3 Methodologies……5

2.3.1 Waterfall model……...5

2.3.2 Evolutionary software development (prototyping)…….....6

2.3.3 Conclusion……..7

2.4 Usability and Accessibility…….....7

2.5 Tools & Appropriate Technologies…….7

2.6 Web Servers……8

2.7 Database…….8

2.8 Server-side programming languages……..9

2.9 Client-side programming languages……9

2.10 Map API……10

2.11 Evaluation…….10

Chapter 3 – Analysis……...11

3.1 Why do systems fail……11

3.2 Analysis of User Requirements……...11

3.3 Feasibility Study……..13

3.4 Alternative and Competitions……..14

Chapter 4- Design…….15

4.1 Database Design……...15

4.1.1 Entity-Relationship (ER) modelling……..15

4.1.2 Normalisation……..16

4.1.3 Database Scheme…….17

4.1.4 Integrity Constraints……17

4.2 System Design……18

4.2.1 Architecture Design…….18

4.2.2 Security Design……19

4.2.3 Functionalities Design…….20

4.2.4 Interface Design……..21

4.2.4.1 Layout design……...21

4.2.4.2 Navigation design…….23

4.2.4.3 Input form design…….24

4.2.4.4 Output report design…….24

4.2.5 Maps Design……25

4.3 Prototyping…….26

Chapter 5 – Implementation…….28

5.1 Database Implementation……...28

5.2 System Implementation…….29

5.2.1 Architecture Implementation……..29

5.2.2 Security Implementation ……29

5.2.3 Programming Implementation……30

5.2.4 System Operations……..33

5.2.4.1 Search and View Pubs…….33

5.2.4.2 Rank the Pub……33

5.2.4.3 Administrator login…….33

5.2.4.4 Maintain Pubs/Users/Default setting records…….34

5.2.5 Interface Implementation……..35

5.2.6 Documentation Implementing……...37

5.3 System Conversion……...38

Chapter 6 – Testing……..39

6.1 Unit Testing……..39

6.2 User Acceptance Testing……..40

6.3 Conclusion……40

Chapter 7 – Evaluation…….....41

7.1 Evaluation Criteria……...41

7.2 Aim and Objectives……..41

7.3 User requirements……42

7.4 Methodology……43

7.5 Schedule……...43

7.6 Technology……...44

7.7 Usability……...44

7.8 Development stages…….45

7.9 Possibly enhancement……..47

7.10 Conclusion…….48

Reference……...49

Appendix A – Self Evaluation……..51

Appendix B – Project Schedule……...53

Appendix C – Requirements Gathering ……54

Appendix D – Three screens shoot of different views of the map ……...56

Appendix E - SQL statements transcript……...58

Appendix F - Programming codes ……63

Appendix G - HTML W3C validation……67

Appendix H - Test Plan……71

Appendix I - Final Files List……..……..74

Appendix J - User Manual…….……..75

Web Information System for a Map of Leeds Pubs

Chapter 1: Introduction

1.1 Project Aim

This project is initially proposed by Leeds CAMRA. The aim of the project is to produce a web based database information system of a map of Leeds pubs. The key objective of this project is to produce a system which will provide information for a list of CAMRA specified ale houses in Leeds in a rich interface map.

This system will allow users to search for specific bars with the map. A content management system for administrators will be produced. Moreover, this developed site will be generic so that other CAMRA branches can easily apply this map system to their own websites.

1.2 Project Objectives

In order to satisfy the aim of the project, the objectives of the project are to

1．Establish the requirements for this system

2．Understand the existing online map of pubs solution

3．Choose the most suitable solution from available technologies

4．Design, implement and test this system

5．Get users feedback

1.3 Deliverables

There are three deliverables produced for this project.

1．A web information system for a map of pubs

2．A report for the web information system

3．A user manual for administrators

1.4 Schedules

Planning is the process of recognizing what work must be completed and how it will be completed within the approved time and resource constraints. Project schedule is the main outcome of the planning process [1]. The deadline of this project is 5pm, 27th of April, 2007. The planed project scheduled will be displayed in the following table.

Planed Schedule

10/10/2006 – 07/12/2006	Background reading
10/10/2006 – 20/10/2006	Aim and minimum requirement
10/10/2006 – 08/12/2006	First iteration
10/10/2006 – 20/10/2006	First iteration user requirements gathering and analysis
21/10/2006 – 01/11/2006	First iteration system design
02/11/2006 – 01/12/2006	First iteration system implementation and deliver
02/12/2006 – 08/12/2006	First iteration system testing and evaluation
09/12/2006 – 09/02/2007	Second iteration
09/12/2006 – 16/12/2006	Second iteration user requirements gathering and analysis
17/12/2006 – 23/12/2006	Second iteration system design
24/12/2006 – 19/01/2007	Exams preparation and exams
20/01/2007 – 01/02/2007	Second iteration feedback gathering and deliver
02/02/2007 – 09/02/2007	Second iteration system testing and evaluation
10/02/2007 – 29/03/2007	Third iteration
10/02/2007 – 20/02/2007	Third iteration feedback gathering and analysis
21/02/2007 – 05/03/2007	Third iteration system design
06/03/2007 – 23/03/2007	Third iteration system implementation and deliver
25/03/2007 – 29/03/2007	Third iteration system testing and evaluation
30/03/2007 – 25/04/2007	System checking and documentations generation
10/03/2007 – 25/04/2007	Writing Report

Milestones

20/10/2006	Submit Aim and minimum requirement form
08/12/2006	Submit Min-project report
18/12/2006	First iteration prototype
09/02/2006	Second iteration prototype
20/03/2006	Progress Meeting
29/03/2006	Third iteration prototype

25/04/2007	Submit the final year report
27/04/2007	Submit the final year report (PDF)

(Note: Gantt chart for the project is available in Appendix B)

1.5 Project Overview

The first chapter gives an overall of the project by demonstrating the aim and minimum requirements. The second chapter analyses the background of the system, as well as containing the illustrations of the problem, user requirements, methodologies and appropriate techniques. The third chapter contains detailed analysis of user requirements and a feasibility study will also be carried out. The fourth chapter shows the design of the database and the system. The fifth chapter explains how to implement the system, and the following chapter explains the testing of the system. The final chapter evaluate the system's functionalities, system's usability and attempts to identify the potential problem.

Chapter 2: Background Research

2.1 Background to the problem

This chapter will report the background literature about finding the solution to the user's problem. It investigates the user's problem and investigates a possible solution using different techniques.

The stakeholder is your customer. The issue of how you and your stakeholder will work together should be established at the very early stage in the project development [1]. In this project, Leeds CAMRA is the stakeholder. "The Campaign for Real Ale (CAMRA) is an independent, non-profit, self-organized organization for real ale, real pubs and consumer rights. CAMRA currently has 80,000 members. CAMRA has over 200 branches throughout the UK. The local branches of CAMRA organize local beer festivals, and run regular events in local pubs and breweries" [2]. Leeds Campaign for Real Ale (Leeds CAMRA) is the local branch of CAMRA based in Leeds. The website for Leeds CAMRA is http://www.leeds-camra.com.

The problem was initially proposed by a visitor to the CAMRA website. Their email stated that the current Leeds CAMRA system has no function to provide information for a list of CAMRA specified ale houses in Leeds in a rich interface map (Appendix C). The aim of this project is therefore to produce a web based database information system of a map of Leeds pubs. Moreover, this map system should be easily implemented by other branches of CAMRA as well. The current Leeds CAMRA website consists of eighteen sections, such as Main, about CAMRA, Festival, Gallery, Download and Members only, yet there is no functionality that provides a method of searching for Leeds ale houses in a graphical map. Further investigations have been done on various branches of CAMRA websites such as, York branch, Scarborough branch, and Bradford branch. The result shows that not one CAMRA website in the UK provides this kind of map service. Therefore, the solution the developer will produce will be useful not only for the Leeds branch but also for any branch of CAMRA in the UK. The potential demand for this map system is enormous.

2.2 User Requirements

The stakeholder of this project is Leeds CAMRA. Tony Jenkins is the project owner representative. Therefore, user requirements for this project are from Tony Jenkins. All requirements are gathering by

meeting conversions and e-mail communications with Tony Jenkins. The general requirements are stated as follows, and the full transcript summary can be found in the Appendix C.

Firstly, the system should be developed using free software and methods as Leeds CAMRA is a non-profit organisation. Secondly, the project should fit into the current host server which is Linux system with PHP 4.3.10-16 and MySQL Ver. 12.22. Thirdly, the system should provide a graphical user interface and the system's data should be stored in persistent database. Fourthly, the system should be able to be accessed from anywhere. Fifthly, visitors will be able to search pubs in the active map by name and postcode. Finally, administrators will be able to maintain records.

2.3 Methodologies

Methodology is defined as a particular procedure or set of procedures [3]. Methodology helps developers meet the budget and schedule during development. Therefore, the use of a methodology is essential.

2.3.1 Waterfall model

There are many methodologies for software development. The waterfall model is the most widely used methodology in the field of software development. The waterfall model is a software development model in which development is seen as a waterfall. The waterfall model divides the system into six stages, which are requirements analysis, design, implementation, testing (validation), integration, and maintenance [4].

The waterfall model will be used to set out the overview of the project and define clear development stages for the project. Moreover, the waterfall model can make the time management and control of this project easier. A strict waterfall model will require all the requirements and analyses fully completed before the start of the implementation, which is very unlikely for this project. The reason is that Leeds CAMRA has never provided mapping functionality before; therefore, users' requirements are not clear at the first stage and the requirements are likely to change during the development process. There are some other problems of using the waterfall model. Firstly, it creates limitations on what you can do and when you can do specific tasks on subsequent stage. Secondly, it is not easy to

notice the failure at the early stage of the development, which increases the risk of the failure of the project. Thirdly, it causes a gap on the understanding of the project between developers and users. And finally, it takes a long time to complete the whole model.

2.3.2 Evolutionary software development (prototyping)

The evolutionary software development, e.g. prototyping, can address the problem of the waterfall model. This model allows developers to develop increasing complete versions of a system over time. This model splits the whole development process into several iterations, and each iteration contains the stages of requirements analysises, designs, implementations, testings (validation), integrations, and maintenances [5].

The evolutionary software development (prototyping) is useful for the situation where requirements are not clear at the first stage or where requirements are likely to be modified in the late development stage [5]. The first requirement received was to build a map system which has a list of ale house in and around Leeds, and allows a user to do perform postcode searches for those ale houses. The problem owner was unable to provide a more detailed requirement in the first meeting. Requirements are added and modified during the development process instead. Therefore, prototyping is suitable for this project. The developer quickly put together a prototype in order to test the system, illustrate ideas, gather early user feedback, and redevelop the system. With the help of the prototyping methodology, users can be involved at the early stage of the system development and can provide useful feedback which will make the system more efficient. Moreover, with users involved, the risk of project failure has been deduced, and it can guarantee making progress every iteration. Moreover, using prototyping model might save development time. For example, the developer spent time to develop a user post comment function and demonstrated it to the problem owner after the first iteration of development. The problem owner decided not to use this function, so the developer abandoned this function in the later development. Lots of time has been saved compared with developing in the Waterfall model.

On the other hand, there are some limitations on applying this methodology, for example, the prototyping development requires more input and feedback from users, which means the developer and users must meet regularly. Furthermore, the prototyping development can be more expensive

because it requires more effort from both developers and users [4].

2.3.3 Conclusion

The waterfall model will be used to set out the clearly overview and stages of the project because the waterfall model can provide a clear development structure. After that, the prototyping will be used to do iteration development because the high possibility of adding requirements and modifying requirements in the later stage of the development.

2.4 Usability and Accessibility

Usability is a fast growing discipline in the field of web system development and should be taken seriously by any professional web developer [1]. Usability means designing a user interface that is effective, efficient, and satisfying [6]. No matter how great the system is functioning, it will ultimately fail if the system is not easy to use. The usability issues will be illustrated by two different techniques; Nielsen's points of usability and Larisa Thomason's Website Usability checklist. The system is developed mainly base on Nielsen Points of Usability and the main checklist is to design a clear and simple navigation system, keep the content clear and simple, support your brand, and provide a facility for visitor feedback as well as testing the site on real users [7].

Accessibility is a subset of usability. Web accessibility means web pages can be presented and interacted with base on users' needs and preferences, and its primary focus is the access by people with disabilities [6]. Accessibility issues will be considered during the development process. Moreover, Lynx, a text only browser, will be used to test the system.

2.5 Tools and Appropriate Technologies

The main tools used for the development of this project are Dreamweaver for the code writing, Top Style for CSS writing, Photoshop for graphic design and MySQL administrator packages for database maintenance. The choice of technologies will be based on the users' requirements as previously discussed in Chapter2.2.

2.6 Web Servers

There are two main web servers available - apache and internet information services (IIS). Apache will be chosen for this system due to three main reasons. Firstly, apache is a platform independent while IIS can only run in windows platform. The system was developed in the developer's machine (WinXP) then move to the CAMRA machine (Linux). Moreover, apache dominates 60.3 percents of the web servers market while IIS only occupies 31.0% of the web servers market [8]. Last but not least, apache web server is free and it is installed on Leeds CAMRA's server.

2.7 Database

There are many different databases available in the market, such as Oracle, MS-SQL Server, MySQL, and PostgreSQL. International Data Corporation claimed that for the worldwide relational database management systems (RDBMS) market for 2005, Oracle sat on the biggest slice, which was 48.9 percent of the $13.8 billion pie. [9] Ms-SQL Server is very popular as well. Based on the research from international data corporation (IDC), Ms-SQL Server is slowly catching up Oracle and IBM; it currently takes 16.8 percent of the relational database management system (RDBMS) market share [10]. It seems that Oracle and Ms-SQL Server are popular. However, both Oracle and Ms-SQL are expensive therefore not suitable for this system because the users' requirements clearly state that all technologies used in this system should be free.

MySQL and PostgreSQL are the most popular free open source relational database management system. Both MySQL and PostgreSQL are suitable for medium size database maintenance, and both have graphical user interface application support. Comparing MySQL with PostgreSQL, MySQL takes the lead in the following point: Firstly, MySQL is more widely used and there is more support available for MySQL. Secondly, MySQL's MyISAM tables are lightweight, so MySQL run faster [11]. In addition, "MySQL is not only the world's most popular open source database but it's also the fastest growing database in the industry." [12]

Due to the better performance of MySQL illustrated above, MySQL has been chosen by this system. Moreover, the Leeds CAMRA's server has been installed MySQL version 12.22.

2.8 Server-side programming languages

There are several web development languages, such as ASP, PHP, and ColdFusion. Active Server Pages (ASP) is Microsoft's server-side language. ASP can generate web pages dynamically. Most ASP pages are written by VBScript, which is easily to understand and use. The limitation for ASP is that it can only be run in windows server with IIS. This clearly doesn't fit the user requirement because Leeds CAMRA website is based on Linux.

ColdFusion is an application server and a software development framework used for the development of the computer software and the dynamic website. The scripting language for ColdFusion is the ColdFusion Markup Language (CFML). ColdFusion has easier syntax than ASP or PHP. It is a platform independent. It supports connections of ODBC with any database systems and it is stable [13]. Moreover, the developer also have better knowledge on ColdFusion due to the developer have industry experience about building commercial systems with ColdFusion last year.

Hypertext Preprocessor (PHP) is a widely-used scripting open source language that is especially suited for Web development and can be embedded into HTML [14]. There is a huge amount of support for PHP. PHP can be developed over any platforms without any charges. The limitation for PHP is that it doesn't support ODBC [14].

From above analysis, we induce that ColdFusion is the perfect choice for this project. However, the basic requirement from users' requirements states that all technologies used in this system should be free, while ColdFusion is not free, therefore, ColdFusion cannot be chosen. PHP has been chosen as the server-side programming language for the development of this project, and PHP 4.3.10 has been setup in Leeds CAMRA's server.

2.9 Client-side programming languages

JavaScript is the most popular client-side programming language. JavaScript can function as both procedural and an object oriented language. JavaScript can be embedded into HTML and run on the client-side. There is not much competitor for JavaScript in client-side programming languages. The choice for this system is therefore clearly JavaScript.

2.10 Map API

There are three main maps API available, namely Google Maps API, Yahoo Maps API and Multimap Maps API. The system will choose using between Google Maps API and Yahoo Maps API because Multimap Maps API provides the most detailed level of street maps than Google Maps API and Yahoo Maps API. However, there is not sufficient support for Multimap Maps API in the Internet. Moreover, Multimap API doesn't support satellite view of the map, which doesn't fit user's requirement.

Google maps API is the most famous maps API, as the Google search dominates 47.40% of the market, while Yahoo's search only occupies 16.00% of the market [15]. Tim Walling says "for Flash developers, the Yahoo! Maps API is better" [16]. Google Maps application is usually implemented by JavaScript with the use of Google Maps API. Although you can develop the Yahoo Maps applications with the choice of flash ActionScript, Flex and JavaScript. It means that the Yahoo maps application can be more dynamic than Google Maps application with the flash. However, Yahoo maps flash application requires the client machine to enable flash to run the application properly. Most browsers don't have flash preinstalled, users have to download the flash plug-in to enable this function, which is unwelcome by a large proportion of users.

In conclusion, the Google maps API will be chosen for this project based on the popularity of the Google search, more widely support, and more detailed level of street maps.

2.11 Evaluation

The waterfall model and prototyping have been chosen as the main methodologies for the development of this project. Apache was chose as the web server, which is running on a Linux server of Leeds CAMRA. The project will be developed based on PHP scripting language with the MySQL database support and Google Maps API.

Chapter 3: Analysis

This chapter will analyse the system using previously discussed techniques. The analysis part is crucial to the success of the project. The main aim of this chapter is to gain a thorough understanding of the user requirements and to outline the system development direction. This chapter will firstly discuss why systems can fail in general. Then, user requirements will be analysed in detail. After that, feasibility study will be carried out to determine the system's viability. Finally, alternative and competitions systems will be investigates to identify improvement opportunities.

3.1 Why do systems fail

It is important to know why systems fail before the development of the system. For example, systems will fail if time and money are underestimated. Moreover, lack of communication might cause the failure of system development. Furthermore, poor functions design and poor interface design might cause the failure of the system.

The system was scheduled with the combination of waterfall model and prototyping model, which can ensure sufficient time and money is allocated to the project. Even if it is not, time and money can be reallocated during the prototyping stage. Regular meetings with stakeholder make sure the system is developed as the user expected, which reduces the risk of project failure. Sufficient amount of time was allocated to the function and interface design, which can be shown from Gantt chart. Moreover, functions and interface are demonstrated to stakeholder as prototype and confirmed by stakeholder. Those can make sure functions and interface are property designed.

3.2 Analysis of User Requirements

This part will focus on the requirements determination step of the analysis phase. The purpose of this part is to turn the high-level requirements into a more precise list of requirements, which are listed as follow.

1. The System will store the following information about administrators: - , login name, password, first name, surname, and administration level.
2. The system will store the following information about pubs: - latitude, longitude, name, postcode,

address, district, website, owned by, run by, phone, rank, picture, and note.

3. The System will store the default latitude, longitude, map level and map's API as default setting.
4. There are two different levels of administration: - Admin and Super Admin, only super admin have authority to create new admin and change the default setting.
5. The system will be built using free techniques. The current server runs Linux, it provides PHP 4.3.10 – 16 and MySQL Ver 12.22.
6. The system should use a persistent data storage.
7. The system can be accessed anywhere.
8. The system should have a graphical user interface.
9. The map in the system should be able to zoom in and zoom out and provide both satellite view and hybrid view.
10. The system will allow users to search pubs from the map by name or postcode.
11. The system will provide a login system which allows administrators to login to the secure pages.
12. The system will allow administrators to create, update and delete pubs information.
13. The system will allow Super Admin to create/edit/delete administrator and create/edit the default map setting.

Based on the analysis of the user requirements list above, there are two types of requirements, which are functional requirements and non-functional requirements. According to the background research section, the main functional requirements for the different levels of users can be illustrated as follows. Visitors can search Pubs in the map by name or postcode and rank the pubs. Admin can login the secure system and maintain (Create/Search/Edit/Delete) pubs details. Super admin can login the secure system, maintain (Create/Search/Edit/Delete) administrators' details and maintain (Create/Search/Edit/Delete) pubs details.

Non-functional requirements contain operational requirements, performance requirements, security requirements and cultural and political requirements [17]. There are three operational requirements for the system. Firstly, this system can be accessed from anywhere. This means this system should be developed as web based application. Secondly, this system will built using free techniques. Therefore, this system cannot use Microsoft package like ASP or any other commercial package like ColdFusion.

The development package for this system is PHP with MySQL based on the previous analysis. Thirdly, this system will allow normal user to search pubs from the map by name or postcode. This requirement could be completed by linking the map with database using Ajax techniques.

There are three performance requirements for the system. Firstly, this system should use persistent data storage. Therefore, the data information shouldn't be stored as text file or XML file, a more persistent database is required, such as MySQL database. Secondly, this system should have a graphical user interface. Therefore, the layout of the interface will be analysed. The most common layouts are table layout and CSS layout. This project will use CSS layout for better usability. Thirdly, this system should be user friendly. Therefore, In order to make sure this system is user friendly, this system will be developed based on GOMS, Nielsen's points of usability and Larisa Thomason's Website Usability checklist. Moreover, accessibility will be considered through the development process as well.

There is one security requirement for the system. User can login the secure pages of the system to create, update and delete information. Therefore, system security should be analysed, such as using MD5 encrypt technique. No cultural or political requirements are anticipated for the system.

3.3 Feasibility Study

Feasibility study is required after user requirements have been defined. Feasibility analysis helps to determine whether a project is worth proceeding and it also identifies the project's risk [17]. The feasibility analysis for this project includes three main parts, which are technical feasibility, economic feasibility, and organisational feasibility. Moreover, legal feasibility will be analysed in the end. The results of those analyses are combined into a Feasibility Study deliverable.

The essence of technical feasibility is used to answer the question of whether the system can be built using current technology [17]. The combination of PHP and MySQL is powerful web application development package, which is sufficient to handle Leeds CAMRA's requirements. Google Map API can provide powerful map server for the system. The system will be hosted by Leeds University Computing School's Linux server, which can provide sufficient disk space and database space. In

short, the system can be built using current technology.

The financial feasibility is used to identify both the financial risk and the costs and benefits associated with the system [17]. Regarding to benefits, the system can provide a useful function for Leeds CAMRA. All the members and visitors can search Leeds pubs from an interactive map in the system. Regarding the costs, the cost of the system is low. There are three main reasons. Firstly, Leeds CAMRA doesn't need to pay for this project as this is a final year project. Secondly, all techniques used in the project are free, such as PHP and MySQL. Thirdly, Leeds University Computing School is willing to host the system free. In summary, benefits of development of the system certainly outweigh its costs, so that the system should be built.

The organisational feasibility is used to identify how well the system ultimately be accepted by its users [17]. The system will not affect other parts of Leeds CAMRA website. Therefore, there is no resistance from Leeds CAMRA. In conclusion, users will use the system when the system completed.

The legal feasibility is used to identify whether the system compliance with legal responsibilities. The system will abide with Data Protection Act, which was introduced in 1984, by using password protection on secure pages. Users have to login to view and modify security data. Moreover, users' password will be encrypted using MD5 technique. Finally, in order to keep the information safe, Leeds CAMRA will backup system data frequently. In summary, the system is built legally.

3.4 Alternative and Competitions

Through an in depth investigation into the systems that exist on the CAMRA pubs server, it was clear that there was no completion or alternative to the system that will be built. There is no single branch of CAMRA provide this map server so far. Moreover, there have not been any previous attempts to build this system before. Therefore, the system is a totally new system. There are some system provides map search of local pubs, such as Google Maps. However, it doesn't satisfy user requirements as user only want to list the CAMRA specify pubs in the system. Moreover, user wants to be able to maintain the pubs' records, which cannot be provided by Google Maps Local Search Server. In short, the system is worth developing and has huge potential demand.

Chapter 4: Design

The design chapter uses the requirements that were gathered during the background research chapter and analysis chapter to create a blueprint for the future system [17]. This chapter will focus on the design of the database and the system. The design of database starts with an initial Entity Relationship (ER) diagram. Database normalisation and integrity constraints follow. Finally, the final database schema is mapped from ER diagram. The System design contains a architecture design, functionalities design, a security design, interface design, and maps design. Then, two prototypes are demonstrated after the database design and the system design.

4.1 Database Design

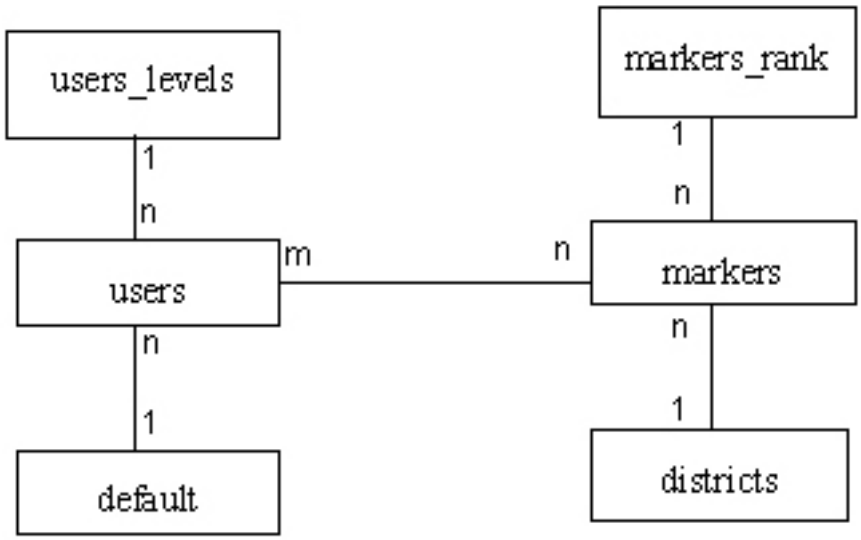
Database Design consists of two steps. The first step is to select the format of the data storage. The second step is to optimise the data storage to perform efficiently [17]. There are two basic types of formats used to store the system data, which are files and databases [17]. Databases storage is chosen over files storage because the user requires data storage persistent. Databases can be object oriented, relational, multidimensional and so on [17]. This system will use relational database as the storage format. Data modelling is the critical activity in the development of relational database so the first part of this section examines data modelling using the Entity-Relationship (ER) model. The second part of the chapter concentrates on optimises database storage efficiency using normalisation. After that, the integrity constraints are checked for data consistency.

4.1.1 Entity-Relationship (ER) modelling

The Entity-Relationship (ER) model is a popular high level conceptual data model [18]. The components of an ER model include entities, attributes, and relationships. An entity is defined as a “thing” exists in the real world. It can be an object with a physical existence or a conceptual existence. [18]. For example, a user and a pub (marker) are the entities for this system. An attribute is used to describe the entity. An entity has a set of attributes and each attribute has a set of permitted values [18]. For example, a user has a name attribute, which normally only permits string values. The relationship represents a relation between two or more entities [18]. There are three different kinds of relationships which are one-to-one (1:1) relationship. There is no one to one relationship in this system. The second relationship is one-to-many (1: N) relationship. For example, one pub (marker)

can be only located in one district but one district can have many pubs. The third relationship is many-to-many (N: M) relationship. For example, a user can maintain many pubs' information and a pub's information can be maintained by many users. The ER diagram for this system was produced based on user requirements as follows.

The ER diagram



There are six tables shown in this ER diagram, which are users, users_levels, markers, markers_rank, districts and default tables. The users table is designed to store user's information, such as log in name and password. The makers table is designed to store pubs' information, such as name, address and postcode. The relationship between users table and markers table is many-to-many (N: M) because a user can maintain many markers and one marker can be maintained by many users. The default table is designed to store system's default setting information, such as map's default latitude, default longitude. The relationship between users table and default is one-to-many (1: N) relationship because there is only one default setting for all users. Users_level table stores what level of security the user has. It has one-to-many relationship with users table. Markers_rank table is used to store rank level of markers and districts table stores districts information in Leeds. Both markers_rank table and district table have one-to-many relationship with markers.

4.1.2 Normalisation

Normalisation is first proposed by Codd in 1972. It is a process of transform data into logical groups in order to avoid redundant data in tables [18]. There are four stages of normalisation, which are First

Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF) and Boycd-Codd Normal Form (BCNF) [18]. This system will use BCNF normalisation to avoid spurious records generation problem and to ensure that each functional dependency is represented by individual relation [18].

4.1.3 Database Scheme

The next step is to map the ER diagram to a schema. Mapping subtypes. There are three different kinds of users which are visitors, administrators and super administrators. There are two ways to map this ER diagram. We could create a user table to store users' detail and a user_level table to store the information of users' level. Otherwise, we could create tables for all the subclasses and the super-class just treated them as ordinary entities, in this case we create four tables which are user, visitor, admin, super_admin tables. The developer chose this first way to model the ER diagram in this project because based on user's requirements this system will not store visitors information, and this system just needs a few administrators. The initial schema is listed as below, table's names are in bold and primary keys are underlined.

users (<u>id</u>, loginname, password, firstname, surname, level_id)

markers (<u>id</u>, latitude, longitude, name, postcode, website, owned_by, address, phone, rank_id, note, district_id, run_by)

default (<u>id</u>, latitude, longitude, map_level, api)

markers_rank (<u>id</u>, rank)

district (<u>id</u>, district_name)

users_levels (<u>id</u>, level)

Foreign key: users(level_id) references users_levels (id)

Foreign key: markers (rank) references markers_rank (id, rank)

Foreign key: markers (district_id) references district (id)

4.1.4 Integrity Constraints

Most database systems have some integrity constraint to follow. Integrity constraints are important to make sure the data consistency [18]. Two types of integrity rules are applied to this system, which are

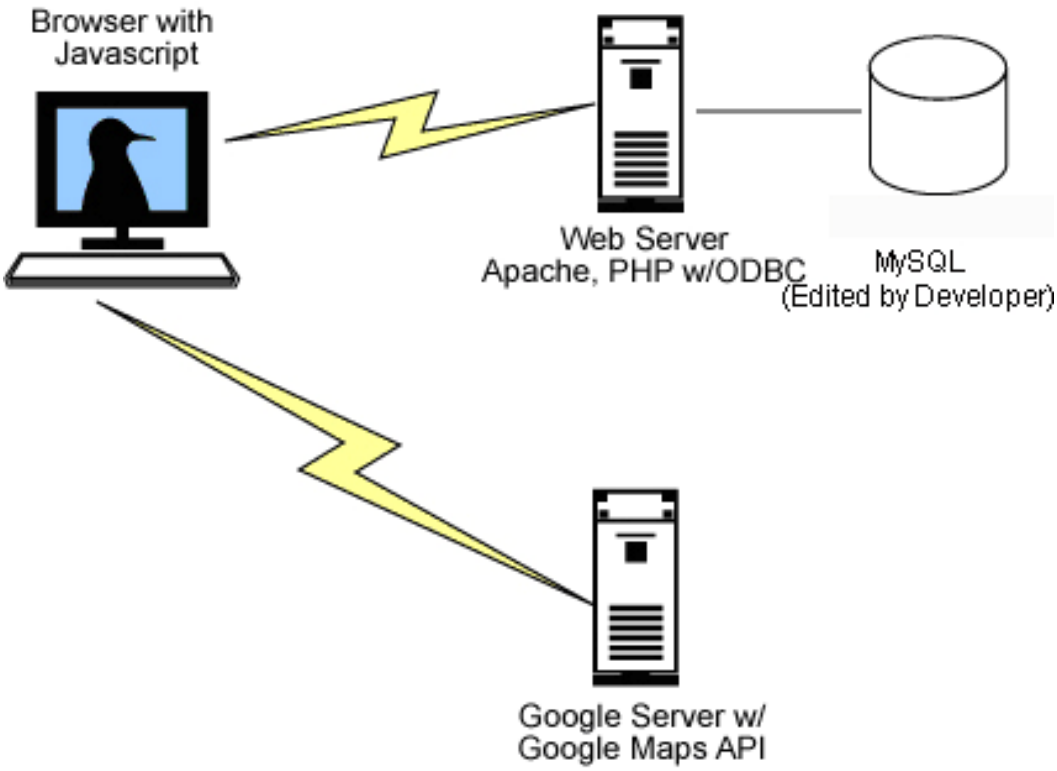
entity integrity and referential integrity. The integrity constraint states that primary key value cannot be null [18]. This is because you cannot distinguish records when both records have null value. Therefore, all the primary keys are defined as no null in the database, such as the id in users table. The referential integrity states that constraint link should be created between two tables in order to maintain the data consistency between those two tables [18]. For example, the users table has a foreign key level_id referencing to the id in users_levels table. This can make sure every records of level_id in users table match with the id in users_levels table. Consequently, it can ensure cascading modified of the record.

4.2 System Design

4.2.1 Architecture Design

The architecture design phase is an important component of the design phase, which describes system's hardware, software and network environment. The architecture design and the hardware and software specification are the main deliverable [17].

This map system is a web-based database system using three-tired architecture. The system will run in the browser with JavaScript on the client computer responsible for presentation logic. This map system will interact with the Web server such as Apache over the Internet responsible for the application logic, with the support of database such as MySQL behind the Web Server responsible for the data access logic and data storage. In addition, this map system will interact with the Google server in order to provide Google Maps API server as well. The architecture design diagram is displayed as follow.



(Source: IBM [19])

The software and the hardware are the major architectural components of the system. The software systems can be divided into four functions, which are the data storage, the data access logic, the application logic and the presentation logic. Hardware systems have three components, which are client computers, servers and the network that connects them [17]. This map system is built to use the existing hardware and software in Leeds CAMRA. The hardware and software specification for this system are Linux machine runs Apache web server, with MySQL database supported and always-on Ethernet.

4.2.2 Security Design

Security design is aiming to protect the information system from the disruption and the data loss, whether caused by an intentional act or a random event [17]. Statistics show that 70% of the unauthorised break-ins, thefts, and sabotage are from hackers external to the organisations in 2001 [17]. Therefore, a system's security design is crucial at present. Five security methods are used in the system, namely user login, MD5 encryption, cookies, concurrency handling and system data backup.

Regarding the user login, the system uses the security login to prevent unauthorised users access and modify data. The techniques of cookies or session variables are used to store authorised users' status. There are three different levels of the security for this system. Security level one requires no permission, which is applied for searching pubs information and ranking pubs. Security level two requires administrator permissions, which are used to maintain pubs information. Security level three requires super administrator permissions, which are used to maintain administrators' information and the system default setting. Regarding MD5 encryption, MD5 is a one way hashing algorithm, which means it is impossible to decrypt it. Users' passwords are encrypted using MD5 technique so that no one can steal your password. Regarding concurrency handling, the update statement and select statement are enclosed into one transaction statement to ensure one record cannot be modified by someone else when you are changing it. Regarding the data backup, the MySQL provides powerful backup functions. Systems' data can either be backup by using MySQL dump or a raw backup.

Two more securities are considered to be applied to this system, which are the record the modification log in the database and the secure shell (SSH). The modified time and person who modified the record can be stored in the time_log and user_log fields in each table by trigger. However, as trigger is not available in MySQL Ver 12.22 therefore this security is abandon. SSL is a communications protocol layer which can be placed between TCP/IP and HTTP suites. It can intercept the Web traffic and provide security between the browser and server. The OpenSSL is a software package from the Open Source Initiative (OSI) to provide SSL server [20]. The software hasn't been installed in the live server and the user doesn't need this level of security as the SSL is more suitable for E-commerce Web system.

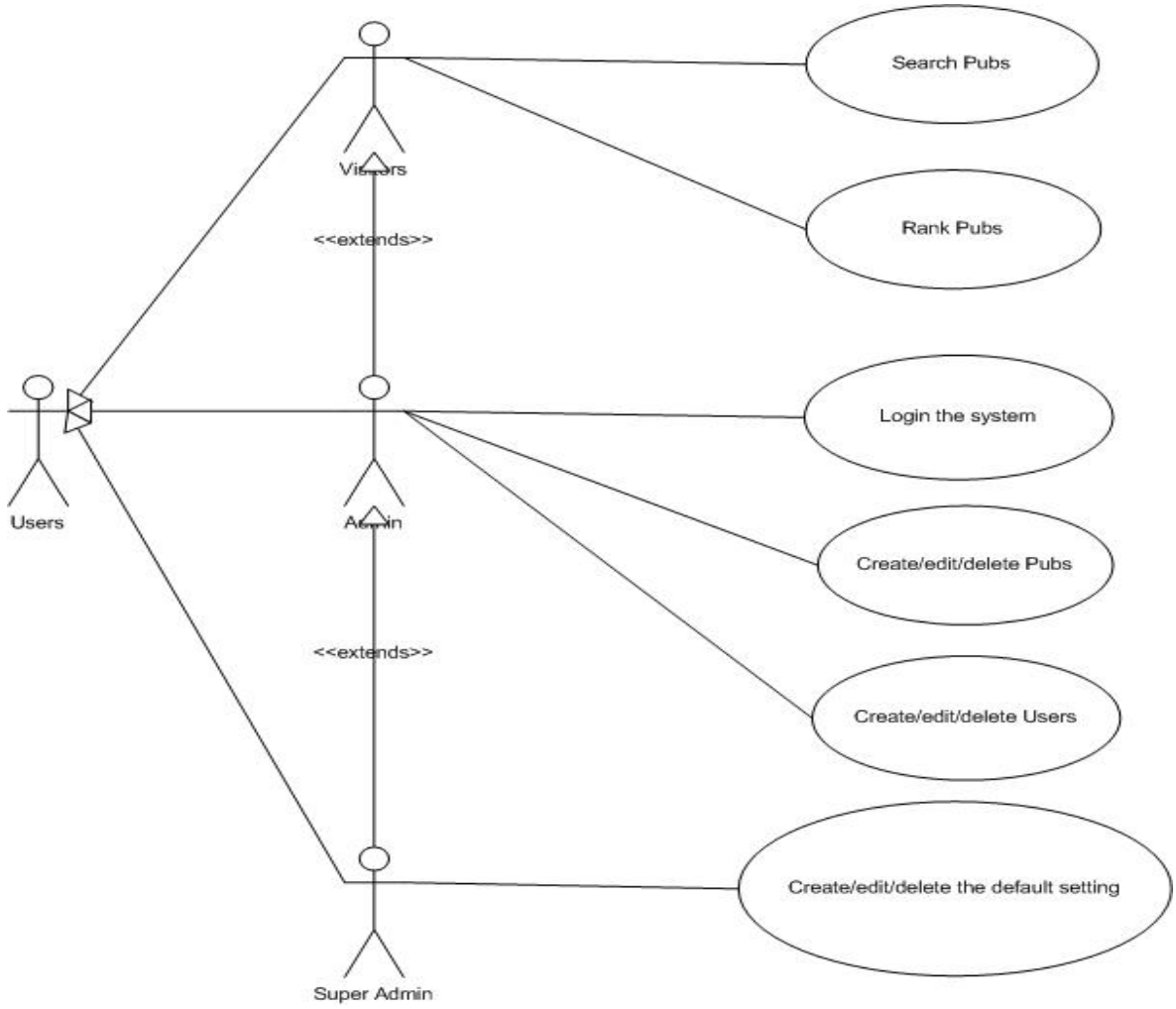
4.2.3 Functionalities Design

Use-Case analysis is carried out to get more detail of requirements definitions. The previous chapter analysed user requirements definitions, whereas this chapter further redefined those requirements into a set of user cases.

The user case diagram is produced for this system as below. There are three different kinds of users, which are visitors, administrators and super administrators. Administrators are specialized users, and

super administrators are specialized administrators, which mean that administrators can do everything what users can do and super administrators can do everything what administrators can do.

UML Diagram: Use Case Diagram



4.2.4 Interface Design

The interface design is the process of defining how the system will interact with users. Users interact with the system through system's interface. It includes the layout, the navigation, forms that capture data, and the reports produced by system [17]. This chapter introduce the principles, the processes of interface design. In addition, usability is important for the interface design.

4.2.4.1 Layout design

Cascading style sheets (CSS) are used in the design of layout to separate presentation and content. There are two ways of implementing style sheets. One way is to embed style sheet in the web page,

the other way is to keep style sheet in a separate file [21]. This system uses linked style sheets instead of embedded styles because linked style sheets are easily maintained.

The layout of the screen is the first element to design. The layout template is produced used multiple layout areas and CSS layout technique. The template is shown as below.

CSS layout

<table>
<tr><td colspan="2">Logo (#header)</td></tr>
<tr><td colspan="2">Title (#gap) Navigation</td></tr>
<tr><td>Map (#mainLeft)</td><td>Text (#mainRight)</td></tr>
<tr><td colspan="2">Footer (#footer)</td></tr>
</table>

The template shows that the layout for this system is divided into five sections. The top section displays the logo and the system header. It is also defined as a #header identity in the CSS. Leeds CAMRA logo is located on the upper left of the page because a logo placed on the upper left offers highest value to the system, which is 84% comparing with 6% on upper right, 6% on the upper center and 4% on other positions [22]. Moreover, the logo should be a link that leads to the homepage in pages except the homepage itself [22]. The title section, which displays a system title and navigation, is defined as a #gap identify in the CSS. The left of main section displays the map and is defined as a #mainLeft identity in the CSS. The right of main section displays text and is defined as a #mainRight identity in the CSS. The bottom displays footer information and is defined as a #footer identity in the CSS. Each section is self-contained so that information in one area does not run into another.

The other important aspect of the layout design is the choice of colours. Readability is important in the colour design. The colour theme for this system is dark because pubs are more popular at night.

This system has six main colours, which are dark green, black, white, grey, yellow and pink. The header and footer section uses dark green colour. The choice of colour is under in-depth consideration. For example, #0055A6, #483D8B and #2E5E00 are examined which is best colour for the header section and the footer section. With developer's analysis and user's preference feedbacks, #2E5E00 (dark green) is selected as the best suitable colour. The gap section uses grey background. This gap section contains a page title and the navigation. The main section, which includes left main and right main sections, has black background. Although 84% of the homepages used black text on a white background and only 4% of the sites used white text on a black background [22], this system uses white text on a black background because the colour theme for this system is dark. Yellow is used for text links and pink is used for visited text links. 74% of the homepages used different colours for links that have been visited and links haven't been visited in order to help users avoid going repeatedly to the same page by mistake [22].

In additional, all interfaces need to include significant white spaces and consistent fonts [17]. Moreover, alt text is used for images in order to improve system's accessibility [23].

4.2.4.2 Navigation design

Navigation is invisible when it's working but it can cause users to give up when there's a problem [23]. The goal of the navigation component of the interface is to make the system as simple as possible to use [17]. The menu is the most command type of the navigation nowadays [17]. Hyperlink menu is used as navigation for this system. The hyperlink menu in public pages provides two choices, which are home and secure page options because those two main choices are sufficient to handle the limit functionalities provided by pubic pages. However, hyperlink menu is not the only navigation provided in public pages. Buttons are used for navigation such as "Back" button to come back to the previous page and "New Search" button to start a new pubs search. The hyperlink menu in secure pages provides three choices, which are public page, secure page and log out. Buttons are used for navigation in secure page as well. Apart from those two navigations, a navigation list is provided and group by similar categories in the secure main page, which listed all available functions for the current user. The list items are grouped by interface objects such as user managements and marker managements rather than interaction actions such as create, edit and delete. In addition, page title used

in the gap section would help people track where they are.

4.2.4.3 Input form design

The aim of the input design is to simply and easily capture accurate information for the system. The inputs design focuses on designing the screens used to enter the information and forms used to provide information [17].

The main input element used in the system is text box and other commonly used types of input element are used in this system as well, such as a drop-down list. Take the pubs search box in homepage as an example, two text boxes with labels and a search button are placed in this search box. The design is displayed below.

All data entered into the system will be validated to ensure the accuracy [17]. Six validation checks are used in this system, completeness check, format check, range check, check digit check, consistency check and database check. JavaScript and regular expression are used to perform completeness check, format check, range check and check digit check. In the above example, a JavaScript validation will be carried out to check whether the postcode data provide by the user is in the correct format, complete and in the proper range when the user clicks the search button. The system consistency is guaranteed by the CSS, while the CSS will be check by W3C CSS validation service. Database check will be performed using PHP.

4.2.4.4 Output report design

Outputs are the most visible of the system, because the primary reason for using the system is to access the information it produces [17]. The goal of output design is output what users want in a clear and simply way [17].

One key issue of output design is the manager information load. The output report should provide information required by users with important information highlighted not all information available. Moreover, as this system is designed for the UK users, the most important information should be presented first in the top left corner of the screen. The output report is broken into two pages, which are the general page and the detail page. The result page for this system displays the general pubs information, which are name, address and postcode. The detail result page displays information about name, latitude, longitude, postcode, website, run by, owned by, address, address, district, phone, rank and note, which are specified required by users. Pub's name is highlighted as users believe it is important to identify a pub.

4.2.5 Maps Design

This system uses Google JavaScript Maps API Version 2, which was launched on April 3, 2006 [24], to provide map servers as previous discussed. The map is presented on the left of main section. This section concentrates on the design of the map.

The relationships between each map's components and how the map works are demonstrated as follow. The browser capacity is checked when the page is loaded. A rich interface active map is presented after user's browser passes the capacity check. This map supports drag, zoom in and zoom out functions. Map control is added to support map pan, and map type control is added to support different views of the map, such as the map view, the satellite view and the hybrid view. Pubs information is retrieve from MySQL database and pubs are placed in the corresponding location as markers in the map. Event listener and click handling are used to display an information window above the marker when a user clicks it. Side panel are presented on the right hand side of the map to display marker's detail information. XML and Asynchronous RPC (Ajax) is used to pass data between the map, side information panel and database. There are around eight hundred pubs in the Leeds area, displaying all pubs in a single map might cause the delay. Marker managers will be used to address this problem. Marker manager can display huge amount of markers on the same map, and it can specify at which zoom levels markers should appear [24].

There are some limitations on the design of this map. Firstly, this design heavily relies on the user having JavaScript enabled in their web browser. If the user does not have JavaScript enabled, the user will be unable to use the mapping service [24]. Secondly, this design doesn't support Geocoding service, which is the process of converting addresses into geographic coordinates. The reason is because the current version of Google Maps API only supports a Geocoding service in The United States, Canada, France, Italy, Germany and Spain [25]. Thirdly, routing is not yet supported by this system as it is not provided in Google Maps API. In summary, the Google Maps API is still in early development, its features are going to change and improve with its popularity. Those limitations mentioned above are hopefully being solved in the near future.

4.3Prototyping

There are two prototypes created in the design stage. The first prototype is aiming to properly design system's functionality. The second prototype is aiming to properly design system's interface and the combination of interface and functionality.

Functions which are stated in the minimum requirements list are designed in the first prototype, such as producing a rich interface map with Leeds Pubs information on it. This prototype have been iteratively developed starting with the design of a home page, and then gradually design the search and login functions in the public page and the content management functions in the secure page. The first prototype was shown to several friends who have lots of web programming experience and redeveloped based on their comments. For example, the function of satellite view support of the system was added based on their comments.

The second prototype was developed based on the analysis of user requirements and usability concerns. System's layout, logo, forms, reports are designed in this prototype. Most of the user interface designs are based on user's preference and requirements, such as what information should be provided in the output reports. This second prototype was shown to stakeholder and iteratively redeveloped based on stakeholder's suggestions. For example, stakeholder preferred the logo contain image and text, rather than text only in the second prototype.

The first prototype and the second prototype are improved to compliance with users' requirements and combined together into a final prototype. This final prototype was demonstrated to stakeholder with the real data in the database rather than test data.

Chapter 5: Implementation

This section outlines how the system was successfully developed, such as programming, testing, and documentation the system. Programming is the largest component of system implementation, the implementation of database and system is illustrated first and is followed by the explanation of the system testing and the documentation. Moreover, this chapter only focuses on the illustration of key features and not every aspect of the system.

5.1 Database Implementation

The system database was created by an SQL file (Appendix E). There were six tables created and each table has a primary key which is a auto-increment integer in order to uniquely identify each row in the table and ensure entity integrity. Three out of six tables, which are users table, markers table, and users_levels table, have foreign key to enforce referential constraint between two tables and ensure referential integrity. Different data types are used in the table, such as Integer type is used for ID and integer number, Varchar type is used for text and Float is used to store latitude and longitude.

The relationship between the tables were defined by add constraint SQL query.

```
ALTER TABLE `camra`.`users` ADD CONSTRAINT `FK_level` FOREIGN KEY `FK_level` (`level_id`)
    REFERENCES `users_levels` (`id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
```

There are two ways to backup the database. The first method is to use the backup utility provided by MySQL Administrator. The other way is to use mysqldump utility. This system will use the mysqldump statement to backup the data. The mysqldump utility is a console-driven executable to backup database to an external source, such as text file. This statement doesn't actually backup the data, while it a set of "Create table" and "Insert into" commands that can be executed by the MySQL server to re-created the database [26].

5.2 System Implementation

5.2.1 Architecture Implementation

Apache web server 2.0.58, PHP scripting engine 5.2.0 and MySQL database server Version 5.04 have been downloaded and installed on the developer's computer which runs Windows XP operation system.

Most of the installation process is automatic, but some configurations are necessary in order to support PHP script and MySQL database by Apache server. There are two main steps to enable the PHP script in Apache server. The php5ts.dll file needs to be copied to the modules folder under apache2 or the system32 folder under windows (WinXP). Then inserted codes "LoadModule php5_module modules/php5apache2.dll" and "AddType application/x-httpd-php.php" into httpd.conf file to enable the PHP module. Moreover, the phpinfo() method can be used to test whether PHP scripting engine working properly in Apache web server. The php.ini must contains "extension=php_mysql.dll" in order to enable the PHP support on MySQL database. MySQL is set to be 3306, default host is localhost, default_user is root and default_password is 123456 in the php.ini file. Moreover, session.save_path has to be set in php.ini file in order to enable session variable to support the function of user login.

5.2.2 Security Implementation

There are five different kinds of security has been implemented, which are user login, MD5 encryption, using cookies, concurrency handling and system data backup.

One of the best ways to protect against unauthorized access to the web system is use a password, therefore, the secure part of the system requires the administrator users have to log in using correct user name and password and users' password. The standard "Message Digest" (MD) function md5 () is implemented to encrypt users' passwords. In the md5("john") = 527bdb5d689e2c32ae974c6229ff785 example, the numeric value on the right hand side represent the digest of the original left hand side text. It is difficult to produce the original text from the numeric result so this is a safety one way function to enhance the security of the system [20]. The md5 encryption is implement to the user table in the database, therefore, $passId = md5 ($passId) is used

when compared the user input password with password in the database. After the user login the secure page, every secure page contains PHP codes to checks whether the user has the correct authority level to view the page. (Appendix F)

A cookie is a small piece of information stored on the client machine by a Web page. The user's username and password are stored in a cookie when they login the system for the first time. The cookie on the machine can determine whether you can the authorization to access the site when user can return the site next time. setCookie () can set cookie information. getCookie () can get cookie value by cookie name. delCookie () can delete cookie by name. delAllCookies () can delete all cookies [20]. (Appendix F)

The problem of concurrency might be raised when the user is writing to the database and the record is being read or write by other users at the same time. Locking is used to solve the concurrency problem. The system will set locks on files with the flock () PHP function to avoid concurrency conflicts. It will lock a record to prevent other users from manipulating the same record at the same time. Moreover, each MySQL table has an associated lock variable. Both read locks and write locks are applied to the system database. In additional, database will backup regularly using the method discussed in chapter 5.1 to enhance the system's security. (Appendix F)

5.2.3 Programming Implementation

The main programming implementation are using the PHP server scripting language to maintain the MySQL database, using JavaScript client scripting language to check and validate data input, and using the Ajax technique to link the Google map with the MySQL database server.

PHP and MySQL is a popular combination. The mysql_connect () method is used to connect to the database server with the user name and password. The mysql_connect_error () method will return an error message saying "Unable to select database" and why it could not connect if the correction fails. The mysql_close () method is used to disconnect the database connection. The database results are output by using the mysql_result () method. The example of connecting the MySQL database using PHP can be found in Appendix F.

JavaScript is used to validate the data completeness, data format, data range etc. For example, the isNaN JavaScript method is used to validate whether the input data for phone number is digit in addMarker.php page. A “Please make sure the phone number is an integer” alert will pop-up if the input data is no digits. The full JavaScript code for the digit validation can be viewed in the Appendix F. The screenshot below is the alert message.

Sometimes the regular expression is used with the JavaScript to validate input data. For example, the UK uses British Standard BS7666 to enforce the format of the UK postcode. It requires all the UK postcode is one of “A9 9AA”, “A99 9AA”, “AA9 9AA”, “AA99 9AA”, “A9A 9AA” or “AA9A 9AA”, where A is an aphbetic character and 9 is a numeric character. The regulation expression for the UK postcode format is “[A-Z]{1,2}[0-9R][0-9A-Z]? [0-9][A-Z-CIKMOV {2}” [15]. JavaScript will check this postcode format using this regular expression when the user submits the form in addMarker.php page. A “Please make sure the postcode format is correct” alert will pop-up as below provided that the input data is in the regular postcode format. The full JavaScript code for the postcode validation is presented in the Appendix F.

There are four main steps to set up a Google maps application. Firstly, the Google Maps API key must be signed up from www.google.com/apis/maps/signup.html before the development of the map system [25]. This system applies two keys one for the local developer machine and the other for the live machine because separate domains must apply for a separate key. Then, the header section of the system should contain a JavaScript to point to the location of the API on Google’s server, and specify the system’s API key as a parameter. Moreover, the body section should contain a div called map, which is the place that Google Map will display. Finally, the code to communicate with Google’s API

is included in the header section by using JavaScript. The full codes to set up this Google maps application can be found in Appendix F.

Google Maps API provides many useful functions to manager the map system and this part will discuss how to implement an initial map system using those functions. The default latitude, default longitude and default start zoom are specified using centerLatitude, centerLongitude and startZoom methods provided by Google Maps API. The GLargeMapControl () method is used to control the pan and zoom of the map. The GMapTypeControl () method is used to change map view types between map view, satellite view and hybrid view. Different views of the map are presented in Appendix D. Moreover, the map system supports a collapsible overview map in the corner of the main map by using GOverviewMapControl () method. Markers information is retrieved from the marker table in the MySQL database and the markers are created using addOverlay method into the map. The click event listener is attached in the marker by using GEvent.addListener () method, therefore, the info window will pop-up with the name of the pub using marker.openInfoWindowHTML (description) syntax when user click the marker [25]. The basic map system with pubs information from MySQL database displayed has been created. The full codes can be found in Appendix F.

The next step is to make the map system more useful and dynamic. Three main improvements have been done which are using the Ajax, using CSS styling of the map and its contents and using event manager to specify which markers should appear in which level. The latitude, longitude and other information from the MySQL database are saved using the GXMLHttp Asynchronous JavaScript and XML (Ajax) object on the client side and PHP on the server so that information are returned to the user without loading the page. An identity called #map is defined in the style.css file to control the visual appearance of the map and a side panel for the map was created to interact with the map. Implementation show that Google Maps API can handle around 800 markers in the map efficiently so there is no need to implement the marker manager into the system. The full codes can be found in Appendix F.

5.2.4 System Operations

5.2.4.1 Search and View Pubs

The picture displays the search pubs function in the public page. Users can search for particular pubs based on pubs' name or pubs' postcode in index.php file. The pubs search function uses SQL like operator to enable comparison to a part of string using % wild card character provided by users. The general results are displayed in the result.php file as in the picture as follows. User can click the name link in order to move to the selected pub in the map. The detail pubs' information is displayed in the detail.php file after the user clicks the detail link, which is as shown in the following picture. The Search code is available in Appendix F.

Name	Boston Exchange
Latitude	53.8154
longitude	-1.5891
Postcode	LS4 2SE
Website	
Run By	
Owned By	
Address	44 St. Anns Lane
District	4
Phone	01132755404
Rank	4
Note	

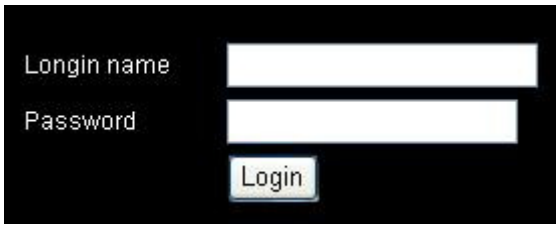
5.2.4.2 Rank the Pub

This function hasn't been produced as the developer ran out of time and it is low priority based on user requirements. However, the attribute to descript pub's ranking was added into the database and a function for administer to maintain pub's rank was provided in the secure page instead.

5.2.4.3 Administrator login

All administrators must login in order to maintain the system data. The JavaScript pop-up alert will be trigger if the user provides a null login name or null password. The user will be redirect to the secure index page using the header ("Location: secure/index.php") code when the username and password are correct while the user will stay in the login page if username or password is incorrect. Administrator can only maintain pubs' records, while the super administrator can maintain administrators' records and maintain the default setting apart from maintain pubs' records. In

additional, the login status is manager by session variables.



5.2.4.4 Maintain Pubs/Users/Default setting records

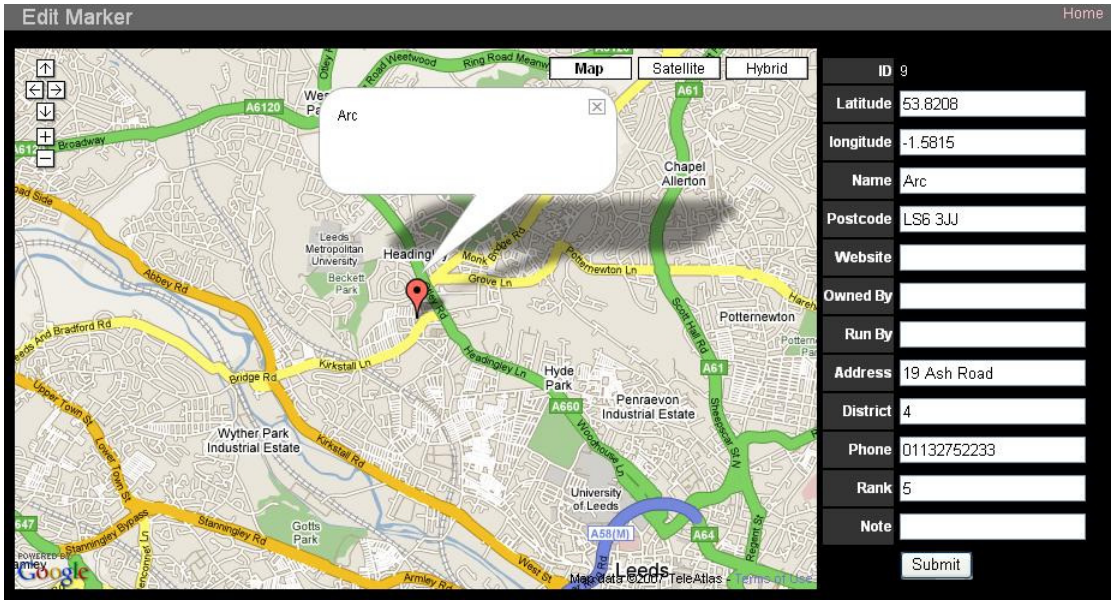
The administrator can search, view, add, edit and delete the records. The confirm box is shown when the user wants to delete the record. The administrative users can search pubs information by providing pub's name or postcode as it is shown in Picture (X) above. The results will be presented in the searchMarkerResults.php page as below.

Results

Home | Se

ID	Latitude	longitude	Name	Postcode	Website	Owned By	Run By	Address	District	Phone	Rank	Note
9	53.8208	-1.5815	Arc	LS6 3JJ				19 Ash Road	4	01132752233	5	
10	53.8222	-1.5781	Arcadia Ale & Wine Bar	LS6 2UE		Market Town Taverns	Market Town Taverns	Arndale Cenre, Otley Road	4	01132745599	4	
11	53.8154	-1.5891	Boston Exchange	LS4 2SE				44 St. Anns Lane	4	01132755404	4	
12	53.8208	-1.577	Box	LS6 2AD				10 Otley Road	4	01132249266	5	
13	53.8203	-1.5785	Headingley Taps	LS6 3HN				North Lane	4	01132200931	5	
14	53.8254	-1.5802	New Inn	LS6 4DL				New Inn	4	01132249131	5	
15	53.8196	-1.5757	Original Oak	LS6 2DG				2 Otley Road	4	01132751322	5	
16	53.8194	-1.5772	Skyrack	LS6 3AW				2 St. Michaels Road	4	01132785836	5	
17	53.8279	-1.5828	Three Horseshoes Hotel	LS16 5JG				98 Otley Road	4	01132757222	5	
18	53.8199	-1.5815	Trio Bar & Grill	LS6 3HU				44 North Lane	4	01132036090	4	
19	53.8279	-1.5828	Woodies	LS16 5JG				104 Otley Road	4	01132784393	5	

Users can click the delete link to delete the record otherwise the user can edit the record by clicking the record's ID, which will direct the user to the editMarker.php page as shown as follows.



The left page will display that particular pub in the map with the pub's name in the info window when the user clicks the marker in the map, and the right hand side will display this pub's detail in editable text box. The user can modify this pub's detail and submit the modification by clicking the submit button.

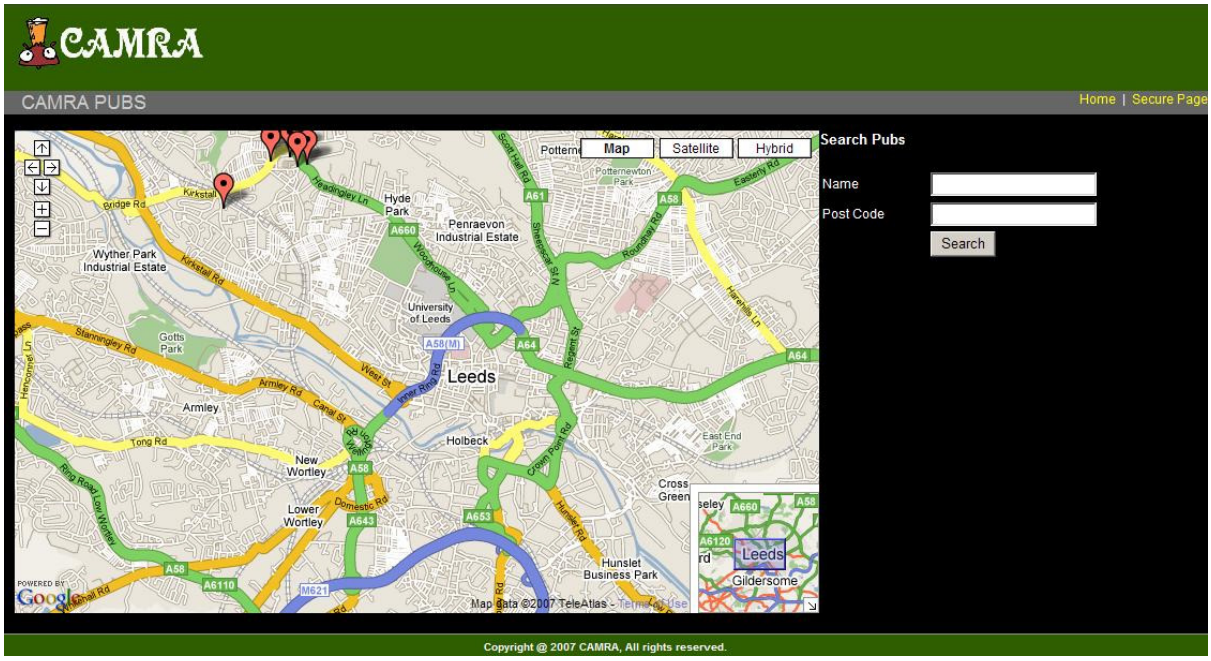
Super administrator can maintain users' records and default setting in the similar fashion as the maintenance of pubs' records. Therefore, the details on users' records and default setting maintenance will not be illustrated here.

5.2.5 Interface Implementation

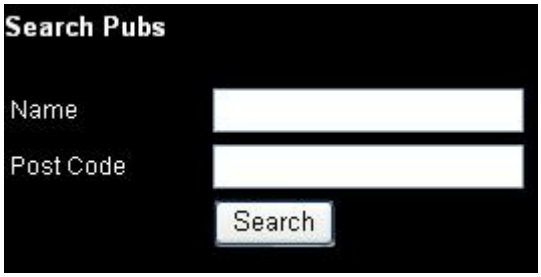
The section deals with the implementation of layout, navigation, forms and reports. Usability is the main focus in interface implementation.

The layout is implemented using cascading style sheet layout as using CSS meets Nielsen 'consistency and standards' usability point, style. The style.css file contains all the style sheet information. The top banner use 100% of the width and 80px of the height, and the background colour is defined as #336600 rather than #2E5E00 as it was previously design in order to complain with Web Safe Palette, which can ensure the system can be viewed from a 256 colour computer system. A 190px width and 50px height logo is located at the top left corner as previous designed using CSS float right

technique. Then the title part is defined as a 100% width, 22px height gap with a #666666 background colour. Both the page title and the menu navigation are displayed in the title section. The main facilities of the system are displayed in the main middle section, which is divided into left main section to display map information and right main section input information and output reports. A min-height attribute has been defined for the main section to ensure this section at least take 450px of height. The left main section takes 68% of the width and has float left attribute, while 30% of the width is allocated to the right main section and has float right attribute. The footer section uses the same colour as the head section and it takes 15px of height. The screen shot for the system layout is presented below.



Navigation menu is simple for this system, which is implemented on the left hand side of title section. The input form is implemented by embed a table into a form, when the user click the search button, the information provided by user will be send to the action page.



The output report is implemented by display information in a table.

Results

Name	Address	Post Code	Detail
Arc	19 Ash Road	LS6 3JJ	Detail
Arcadia Ale & Wine Bar	Arndale Cenre, Otley Road	LS6 2UE	Detail
Boston Exchange	44 St. Anns Lane	LS4 2SE	Detail
Box	10 Otley Road	LS6 2AD	Detail
Headingley Taps	North Lane	LS6 3HN	Detail
New Inn	New Inn	LS6 4DL	Detail
Original Oak	2 Otley Road	LS6 2DG	Detail
Skyrack	2 St. Michaels Road	LS6 3AW	Detail
Three Horseshoes Hotel	98 Otley Road	LS16 5JG	Detail
Trio Bar & Grill	44 North Lane	LS6 3HU	Detail
Woodies	104 Otley Road	LS16 5JG	Detail

After the HTML and CSS system layout have been completed, the W3C markup validation service was used to validate the page based on XHTML 1.0 Strict standard. There were eight errors occur when the index.php was tested, such as there is no attribute “width” in table, character “&” is the first charter of a delimiter but occurred as data. All the errors were corrected and the index.php page was tested again with W3C markup validation sever and it successfully pass the test, which enhance system’s accessibility and usability. The detail can be found in Appendix G.

5.2.6 Documentation Implementing

There are two main types of documentation generated to help programmers, system analysts and users to understand the application software and maintain the system, which are system documentation and user documentation, such as user manual [17].

A user manual is produced for the system (Appendix J). The reason why a paper based manual is chosen rather than online manual is because paper based manual is simpler to use as it is more familiar to users, especially novices who have less computer experience [17]. Moreover, this paper based manual can be downloaded from the Internet. This user manual is produced during the development process rather than in the end of the project. Moreover, the time for user manual producing is recorded in the scheduled Gantt chart.

5.3 System Conversion

System conversion is the way users switched between the old and new systems. There are two different type of conversion, one is direct conversion and the one is parallel conversion [17]. Direct conversion is used to the system from local developer machine to the University of Leeds server when it was completed in order to get efficient feedback from users and testing. The new server provided by University of Leeds supports Apache web server, PHP 4.3.10-16 scripting language and MySQL Ver 12.22 on the UNIX machine. The file storage server is wwwdev.comp.leeds.ac.uk and the database host is csdbdev.leeds.ac.uk. Files can be uploaded via FTP to the server via ftp.csunix.leeds.ac.uk. PHP is not enabled by default. The ".htaccess" file is created with contains of "AddType application/x-httpd-php.php" to enable PHP support. Moreover, the "htaccess" contains "Directory index.php" to load the index.php automatically. A new Google Maps API key is applied to the new server directory. The final files structure can be viewed in Appendix I. The system can be accessed via http://wwwdev.comp.leeds.ac.uk/efy2jz/CAMRA/index.php address. This address is not accessible from outside university network, however in order to get wider feedback from different users, an access by outside university network authority to this address has been given by university IT support group. The system will stay in the location at this stage based on user's requirement as the Leeds CAMRA website is hosted by University of Leeds as well.

Chapter 6: Testing

Software must be tested to correct the errors before delivery to the stakeholder when source code has been generated [27]. System testing is a critical element of system development, which represents the ultimate review of specification, design, and code generation [27]. The developer will exercise the internal logic of system components and exercise the input and output of the system following the systematically testing structure and using disciplined techniques to correct errors in program functions, behaviour and performance. There are many tests available, but this project will use two, which are unit testing and user acceptance testing. Unit testing will test individual programs units such as functions or objects to insure that they work properly. User acceptance testing is used to confirm that system is complete, meets user requirements and is acceptable to the users after the delivery of the system [17].

6.1 Unit Testing

The developer carried out the Unit testing by examining the smallest unit of the system, such as the system component or module to ensure that the module or program performs its function as defined in the program specification. Two different kinds of unit testing approaches have been used to test the system, which are black-box and white-box. The developer tested unit to check whether it meets the requirements stated in the program specification to perform the black-box unit test and then look inside the program to test the system's major elements, such as edit marker function, to implement the white-box unit testing.

The main units have been tested are system architecture, system security, system database, system functions, system layout, system navigation, system input form, system output report and map functions. Many bugs were found in the system during the unit testing. The developer had corrected all the errors found during the unit testing and uploaded them to the server to retest them. The detail unit testing plan which defined a series of tests that will conducted and the corrected results can be found in the Appendix H.

6.2 User Acceptance Testing

The users with support of the developer carried out the user acceptance testing at the final stage of the system development to confirm that the system is complete and is acceptable by the users. This required a significant amount of the users' involvement and the data used for the user acceptance testing is live data.

A user acceptance test is done in two stages: - alpha and beta testing. The Alpha tests often repeat previous tests using made-up data but are conducted by users rather than the developer to ensure they accept the system. Users closely test the system with real data. There are some problems that need to be investigated during the user acceptance testing. For example, the user complained that there is no an alert message pop-up to ask for confirmation when user deleted the record so that user might easily delete the record by mistake. This feature was immediately added to the system. The user required that the latitude and longitude should be displayed on the side panel of the map when the user clicks the map. However, this function hasn't been implemented in the end as the developer ran out of time and this function is low priority. The level of priority on each individual requirement is presented in the test plan. The detail user acceptance testing plan, which can be found in Appendix H, was produced to illustrated whether users think this system satisfy their requirements.

6.3 Conclusion

The unit testing and user acceptance testing found many errors. Most of the errors that were found have been corrected, however there are still some remaining errors due to lack of time. In general, it can be concluded from the testing results that this system is successfully completed with both the developer and Leeds CAMRA satisfied.

Chapter 7: Evaluation

This chapter will evaluate the system's functionalities, system's usability and identify system's problems. Possible future enhancements will be discussed. What's worth mention here is that there is no need to evaluate the alternative and competition for the system as previous discussed in the analysis chapter there is no alternative and competition to the system.

7.1 Evaluation Criteria

Judge whether the system is success based evaluation on the aim and objectives, user requirements, possibly enhancement, methodology, schedule, technology, usability, and the development stages (analysis, design, implementation, testing).

7. 2 Aim and Objectives

The overall aim for the web information system for a map of Leeds pubs is met. A web based database information system of a map of Leeds pubs has been successfully built using PHP, MySQL, JavaScript, Ajax and Google Maps API, which is the project aim. The user can search pubs information from the map in the public index.php page. The administrative users can login the system using the content management system to maintain the records. This system was built using the Nielsen's point of usability and the website usability checklist, which ensure that the system was built generically so that other branches can easily implement this system to their local websites.

The overall objectives for the web information system for a map of Leeds pubs are met. User requirements are established by a series of meeting with stakeholder during the development process. Then the existing online maps of pubs solutions are investigated and it turns out only a few solutions exist, such as Google Maps Local Servers. However, none of the existing solutions can meet Leeds CAMRA's requirements. After that, the available techniques and methodologies to build the system are analysed and the choice of techniques and methodologies are justified. The system was analysed, designed, implemented and tested using Waterfall methodology and prototyping technique. The system was produced and available online. Finally, users' feedbacks are gathered and the system was redeveloped based on users' feedbacks.

7.3 User requirements

The user requirements is a criteria to measure the functionality of the system, it is also a tradition way of evaluating a system. The user requirements for the web information system for a map of Leeds pubs are met.

The system must satisfy the four minimum requirements for the system. The rich interface map provided by Google Maps API with Leeds pubs information on it has been created. This map support move, zoom in and zoom out functions using GLargeMapControl () method by Google Maps API and the GMapTypeControl () method support the satellite view of the map. Then, the public main page provided the advanced search function for Leeds pubs. After that, a login system for administrators has been developed. The user's password has been encrypted using md5 () method. Finally, the developer produces a basic content management system for administrators, which provide functions to maintain users' records, pubs' records.

Some of the possible extensions for the system have been produced. Extra advanced functions have been developed, such as function to change system's default setting. Moreover, system's security has been enhanced by using cookies to store user information, and using lock function to handle concurrency problems. In addition, a user manual was produced for this system. There are some possible extensions haven't been developed due to lack of time, such as the ranking function for users to rank the pubs. Those functions might be implemented in the later version of development.

Other main user requirements are satisfied, the evaluation of those user requirements are displayed as below. Firstly, this system can be accessed anywhere. This has been achieved by building the system as a web-based system. Secondly, this system will be built using free techniques. This was achieved as this system was built using Apache for server host, PHP for server side programming, JavaScript for client side programming, MySQL for database storage and Google Maps API for providing map server, which are all free techniques. Thirdly, this system should have a user friendly graphical user interface. This was achieved by developing the system using CSS layout and based on Nielsen's point of usability and Website usability checklist. Fourthly, this system should use the persistent data storage. This was achieved by choosing MySQL as the database server as MySQL is very persistent.

Finally, this system should have two different levels of admin users. This has been achieved by putting admin and super admin level into the users_levels table. The login function in the system will check which level the administrator is and give corresponding levels of authority to the user.

7.4 Methodology

The evaluation on how suitable the methodology was adopted for the system is critical to measure the success of the system development. It appears that waterfall model and prototyping are suitable for this system development as it guides the developer through the whole development process. The waterfall model presented the overview structure of the system development while the prototyping ensure the development ran smoothly and iteratively. Problems can be identified in the early stage on development using the prototyping, which can save time and money of the development. For example, the system initially would implement a function which allowed visitors leave comment for the pubs. However, it turned out that the stakeholder didn't want this function at all during the system feedback gathering in the first iteration of development, therefore, this function was abandoned and time and money were saved. Other people claims that the prototyping is not a sufficient methodology as it involved too much input from users and it wasted the developer's time by coming back to the same point over and over again. However, the use of prototyping did guarantee the quality of the system and it save time and money in the final testing and bug fixing stages as fewer errors are found in the final system testing. Waterfall model and prototyping model are proved to be the correct methodology for the development of this system as they success guided the system development.

7.5 Schedule

The initial estimate schedule is optimistic even all eventualities have been considered as this project involved some advanced techniques such as Ajax technique. The other reason is that the developer underestimated this project because this system didn't involved complex database design. However, it turned out the development of the system is more complex than what the developer expected. Some functions haven't been implemented simply because of running out of time. Moreover, another mistake was made when designed the initial schedule which is that the Easter holiday was ignored. This mistake caused the delay of the third iteration and feedback gathering as the stakeholder are in vacation. Moreover, the initial schedule arranged too much time on system development rather than

document generation and project writing. In summary, the system was successfully completed in time even it didn't follow exactly as the plan in the initial schedule. More details about project scheduling can be found in the initial schedule Gantt chart from the Appendix B.

7.6 Technology

The main technologies used in the system development are Apache, CSS, PHP, MySQL, JavaScript and Google Maps API. It can be concluded that the choice of techniques are correct after the complete of the system. The Apache is platform independent therefore it can stably host the system both on the developer's machine and on the live machine. The layout is consistence and easy maintained by using CSS layout rather than the table layout. The PHP server side scripting language worked perfectly well with the MySQL database server, they are sufficient enough to provide all the functions required in the user requirements. They perform well in enhance the system's security as well because the PHP can provide md5 () for encryption and the MySQL can provide easily backup mechanism. The JavaScript utilises a value to the functions in the client side such as it pop up a confirmation window when the user want to delete a record. The Google Maps API also provides good server on mapping functions, for example, it provide a map with zoom in and zoom out control. There are possible alternatives to the above technology but the above choice of techniques are the most suitable technology based on the analysis in the background research section and it was proved by the successful development of the system.

The system overall is successful developed using the above techniques. However, there are some technological problems remain after the development. For example, the min-height CSS bug hasn't been fixed, it only enforced the minimum height for the main body as 450px by using Firefox browser, but it didn't work in the IE browser. Moreover, the MySQL server doesn't support trigger under MySQL4.0, therefore, the information about who modified the record and when is difficult to implemented. Furthermore, the stored procedure can improve data handling speed significantly, but it is not support in MySQL4.0 yet.

7.7 Usability

The usability is crucial as an unfriendly system is likely to fail by user resistance. This system was

developed using Nielsen's point of usability and Website usability checklist which can ensure the system's usability.

There are many features to improve the system's usability. For example, the Google Maps API code was initially stored in each individual PHP files. Then the developer decided to move the API code from the PHP files into a single api.php file and using <?php include 'apikey.php'; ?> statement to call the Maps API when it is needed. The API code can be modified by edit only one file which is api.php in this situation. Finally, the Maps API was stored into the default table in the database and a function to modify the Maps API based on graphic interface was developed. This improved the system's usability as the users could modify Maps API easier. Another example is that a delete confirmation box is implemented, which can reduce the chance of delete the records by mistake. One more example is that the CSS is separated from PHP code. All the CSS are saved in the style.css file, and using a external link to each PHP file. This enhances the usability on maintaining the layout and appearance of the system as all styling information is stored in one single file namely style.css.

7.8 Development stages

Every development stages are important components of the system development. Therefore, this part will evaluate each individual development stage: analysis, design, implementation, and testing.

Although the possibly causes of system failures are presented in the analysis stage, the developer still made some mistakes which are in the list. For example, the developer underestimated the system which caused chaos and panic during the development when the system cannot be developed as it was planned in the initial schedule. However, the developer spent more effort in the late development stage, which helped to avoid the failure of the system. The high-level user requirements were analysed into a solid precise list in this stage which contained functional requirements and non-functional requirements. In conclusion, the analysis stage successfully outlined the development direction, and clarified what is supposed to be in the system and whether it was feasible.

The database and system are successfully designed in the design stage. The developer was too ambitious to design the system at first. The developer attempted to create lots of functions which were

not required by the stakeholder, such as post comment, add to favourite, share the pub, automatic install etc. This design made the database and system's functions extremely complex, and plenty of time has been wasted on meaningless unwanted functions during the first iteration of development. However, with the help of prototyping this serious mistake has been found in the feedback collection stage of the first iteration. All the unwanted designs were abandoned in order to avoid system failure. Communication with the user exerted a value in this situation. Finally, the database was well designed with eight normalisation tables in the BCNF form, and the system's architecture, security, functions, interface are designed to an acceptable level with the help of two prototypes.

The implementation stage is the most difficult part of the system development. There are many difficulties incurred in this stage. Firstly, the Geocoding service cannot be implemented into the system because it is not available to the UK in the current version of Google Maps API. However, a button which pop-up a Multimap web page was provided. User can find the latitude and longitude of the corresponding address through the pop-up Multimap page, which issue the Geocoding problem. Secondly, there is not free UK postcode to latitude and longitude database available. The post office owns it and sells it to various companies. As this system use free techniques only, therefore, the developer will not purchase this database from the post office. The developer created his own postcode to latitude and longitude database instead. Although this is not as precise as the database from the post office, it is precise enough for the system. Thirdly, no solution have been found for the JavaScript disabled browser as Google's Map API relies heavily on JavaScript, and this is the only option for Google Map's. Fortunately, most browsers support JavaScript. Therefore, this problem would not cause the system failure although it did affect system's accessibility. To summary, the implementation of the system was success as it met user's requirements and is working at a reasonable level.

The testing stage was also successfully carried out. The unit testing was carried out during the development process and the user acceptance test was carried out in the end of the development. Although there are a number of errors found during the test, all the errors are fixed and both the developer and the Leeds CAMRA agreed that this system worked in a satisfactory level.

7.9 Possibly enhancement

Many possibly enhancements can be added to the existing system in order to enhance system's functionality, usability and so on. The browser in mobile phones can request a Wireless application Protocol (WAP) page from a WAP site on the Mobile Internet. WAP page are usually written in Wireless Markup Language (WML) [20]. Therefore, the system could be made available through mobile phone using WAP and WML. The most significant benefit of implementing this feature is that the user can use their mobile search pubs information when the user is on his way to the pub. Moreover, the system can added a GPS location feature when the system has been built using WAP so that the user can drive to the pub with GPS routing through the system. However, those features are too time consuming, therefore, it is unlikely to develop those features in the current version of the system.

All the red markers on the map are meaningless. The red marker can be replaced by customised image such as a beer image by using Google's GIcon object. However, this feature hasn't been added as this is not required in the user requirement list. It is possibly to implement this feature when it is needed to enhance the system's usability.

With the current system it would be possible to implement a function that could search all the pubs in the UK rather than in Leeds. However, the data collections for the UK pubs are too much work for this report. Therefore, the current version of the system only support pubs search in Leeds. The later version might cover other UK cities. Otherwise, it might be easier to installed the current system to other branches local CAMRA websites, and then to personalise the system and data.

The manual was produced and it was available to download in the current system. However, online help function is urgent to produce when new complex version of this system was released, which contains the possible enhancement described above. The online help wasn't produced as it is not essential in the current version of the system.

7.10 Conclusion

The project aims and objectives are fully met. The functionalities implemented for the system met most of the user requirements. However, there was one function in the user requirements list hasn't been implemented which ranks pubs. This was not implemented due to lack of time. The user requirements are met due to ranking pubs is a low priority function. The correct methodologies are used for the system development, namely waterfall and prototyping model. The initial schedule guided the development process efficiently even it was a bit optimistic. The techniques used in the system development are the most suitable techniques based on the in depth analysis in background research chapter. The usability of the system was in a high standard as the system developed based on the Nielsen's point of usability and the Larisa Thomason's Website usability checklist. There were problems and difficulties that occurred in every stage of system development. However, every development stage was successfully completed. There is room for possible enhancement, which offers opportunity for future improvement.

The development of the system has exercised and extended the developer's academic skills on information systems and database development software engineering and web programming as previous discussed in design chapter and implementation chapter. It also has exercised and extended the developer's professional and transferable skills, such as project scheduling (Gantt chart), background research and analysis (user requirements), project management and organisation (methodology), and project demonstration and reporting (writing report and manual).

It might be too early to say the system is success at this stage as it takes a few months to get comprehensive user feedback. The evaluation carried out above and Leeds CAMRA's feedback indicated that the system is overall success despite that there are unfinished task remained and rooms for improvement.

References

[1] Thomas J. Shelford, Gregory A. Remillard (2002), **Real Web Project Management**, Addison-Wesley, Boston

[2] Campaigns for Real Ale, http://www.camra.org.uk/page.aspx?o=about (Accessed on 02/11/2006)

[3] Creswell, J. (2003). **Research Design: Qualitative, Quantitative, and Mixed Methods Approaches**. Thousand Oaks, California: Sage Publications

[4] Avison, D; Fitzgerald, G (2003), **Information Systems Development,** Third Edition, McGraw-Hill

[5] Haag, Stephen; Maeve Cummings; Donald J. McCubbrey; Alain Pinsonneault; and Richard Donovan (2006), **Management Information Systerms: For the Information Age**. 3rd Canadian ed. New York: McGraw-Hill Ryerson

[6] Jim Thatcher, Paul Bohman, etc (2002), **Accessible Web Sites**, Apress, New York

[7] Website usability checklist, http://www.netmechanic.com/news/vol7/design_no4.htm (Accessed on 10/12/2006)

[8] November 2006 Web Server Survey, http://news.netcraft.com/archives/web_server_survey.html (Accessed on 24/11/2006).

[9] Completive analysis – worldwide RDBMS 2005 Vender Shares, http://www.oracle.com/corporate/analyst/reports/infrastructure/dbms/idc-201692.pdf (Accessed on 10/11/2006)

[10] SQL Server‘s market share continue to grow; MySQL could spark big changes, http://arstechnica.com/journals/microsoft.ars/2006/5/31/4163 (Accessed on 24/11/2006)

[11] SQL Server‘s market share continue to grow; MySQL could spark big changes, http://arstechnica.com/journals/microsoft.ars/2006/5/31/4163 (Accessed on 24/11/2006)

[12] Market share, http://www.mysql.com/why-mysql/marketshare/ (Accessed on 24/11/2006)

[13] Arman Danesh etc, **Mastering Coldfusion MX** (2002), Sybex, San Francisco and London

[14] What is PHP? http://www.php.net/ (Accessed on 24/11/2006)

[15] Google, Yahoo! And MSN: Property Size-up, http://weblogs.hitwise.com/bill-tancer/2006/05/google_yahoo_and_msn_property.html (Accessed on 24/11/2006)

[16] Why the Yahoo! Maps API is Better Than Brand X Maps API,

http://mcmanus.typepad.com/grind/2005/06/why_the_yahoo_m.html (Accessed on 24/11/2006)

[17] Alal Dennis, Barbara Wixom (2003), **Systems analysis design**, second edition, John Wiley & Sons, New York

[18] Ramez Elmasri, Shamkant Navathe (2003), **Fundamentals of Database Systems**, Fourth Edition, Addison Wesley

[19] DB2 and Open Source: Put yourself on the map with Google Maps API, DB2/Informix, and PHP on Linux, http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0602lurie/index.html?S_TACT=105AGX52&S_CMP=cn-a-db2 (Accessed on 10/11/2006)

[20] P.K. Yuen, V. Lau (2003), **Practical Web Technologies**, Pearson Education Ltd

[21] Jakob Nielsen (2000), **Designing Web Usability: The Practice of Simplicity**, New Riders.

[22] Jakob Nielsen, Marie Tahir (2002), **Homepage usability 50 websites deconstructed**, New Riders.

[23] Jared Spool (1999), **Web Site Usability: A Designer's Guide**, Morgan Kaufmann Publishers.

[24] Google Maps API, http://www.google.com/apis/maps/documentation/ (Accessed on 10/01/2007)

[25] Micheal Purvis, Jeffrey Sambells, Cameron Turner (2006), **Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional**, Apress, New York.

[26] Diary of A Webmaster, part 4, Backing up with MySQLDump, http://www.sitepoint.com/article/backing-up-mysqldump (Accessed on 01/02/2007)

[27] Roger S. Pressman (2000), **Software Engineering: A Practitioner's Approach**, fifth edition, McGraw-Hill.

Appendix A - Self Evaluation

I have great interest in web information systems development, therefore, I am glad that I have had the chance to develop this system although it was stressful and suffered problems during the development process. My knowledge and techniques skills, such as PHP, MySQL, JavaScript and Google Maps API, are greatly improved, and I gained valuable experience on project management skills, such as Gantt chart. Moreover, the project also provided me a chance to apply the knowledge I learnt from the University into practice. The project involves the information systems and database development software engineering and web programming. Introduction to Information Systems (IS11) provides the developer with knowledge of information system development such as methodology and project management. Introduction to Databases (DB11) and Database Principles and Practice (DB21) has taught the developer how to develop databases, such as ER modeling and normalisation. Introduction to Programming (SE15), Object-Oriented Software Engineering (SE20) and Practical Software Development (SE24) has given the developer software engineering knowledge such as producing UML diagrams. Web programming can be learned from Introduction to Computer Systems (SY11) and Internet Systems Technologies (SY23), such as Apache web server configuration, PHP programming and CSS layout. Moreover, new technology and knowledge will be learnt during the development process, such as utilising Ajax techniques.

The important lessons I have learnt in this report are listed below:

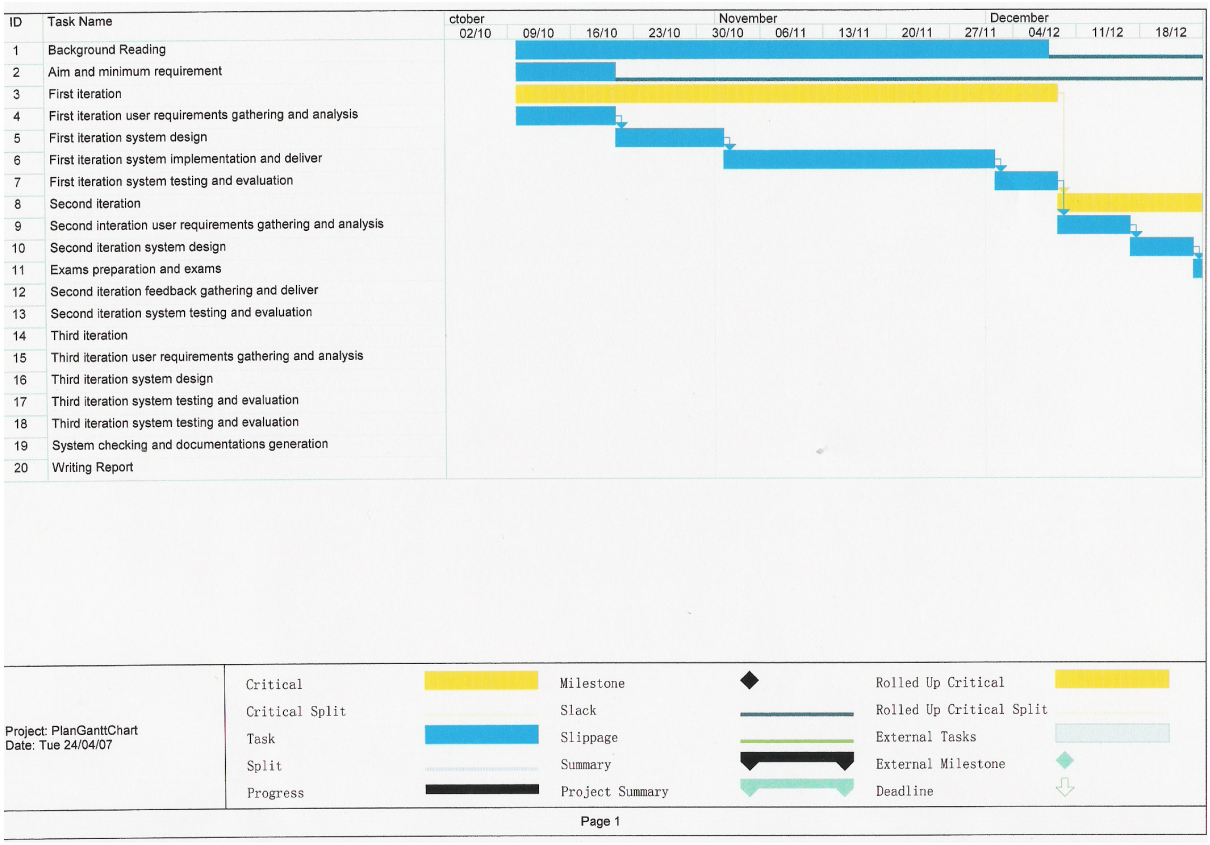
1. I should never underestimate the project. I judged that the system is too simple compared with the system I haven't built during my placement. I even proposed a new system to develop and met with my supervisor and assessor to discuss the feasibility of the new system. The new proposed system was abandoned because nobody wants that system, and it is too complex to be done with limited time constraints. Plenty of time was wasted during this redevelopment process.
2. I should produce a more efficient plan for the system development and follow it. I allocated too much time on functionalities development rather than document and report generation. I have to spend so much time and effort on writing the report in the last stage of the development. It made me feel stressed and under pressure. Moreover, I have lacked the time to complete an in-depth user feedback collection and testing. Therefore the final redeveloped is poor. Fortunately, a number of tests have been done by the user during the development and a simple test was

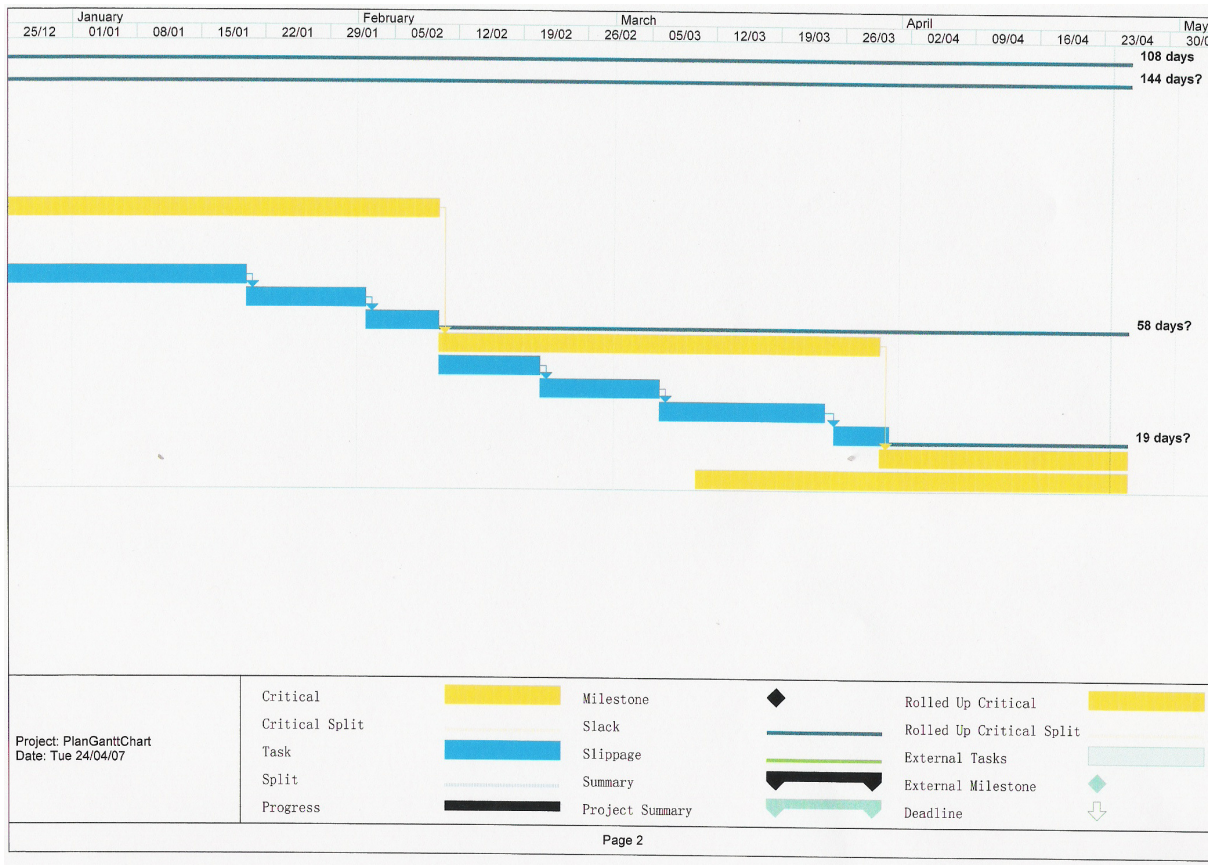
performed by the user at the last stage of development. My test plan was the combination of different tests by the user rather than a systematic test by the user simply because of running out of time. I should have started to write the report earlier so that I could have had the draft version checked by my supervisor so that I can revise it.

3. I should have more communication with the user (problem owner). Most of the user requirements are collected from the third party was by informal methods, such as emails and meetings. I had to spend hours and hours organising those materials. The formal user requirements collection methods, such as interview, should be adopted next time so that material collected are more valuable and well organised. Moreover, most of the communications with the user are completed at the start of the development and at the end of the development. More communication with the user should be carried out during the development process so that problems can be solved immediately.
4. I should have more contact with my supervisor and my assessor. They can provide efficient guide to the development process. I should have prepared questions before the meeting with the Supervisor, and put more efford on mid-project report and progress meeting as those are the main opportunity to get feedback from assessor.
5. The writing style and grammar should be checked by chapter rather than have somebody check it through in the end. This is particular important for international students. As I put report writing in the last stage of the development, therefore, there is no much time left for asking other people checking my report chapter by chapter. Fortunately, one of my colleagues from my last year placement company is willing to spend his whole weekend on checking through my report before the deadline. However, there certainly are a small number of grammar mistakes in the report as not sufficient time is allocated to the grammar checking.

If I have the chance to develop the project again, I would not underestimate the project and schedule the plan too optimistic. I would also make sure sufficient time was spent on report writing with someone checking through the grammar mistakes in the middle of system development so that there would be sufficient time for revising the report and redeveloping the system. Moreover, I would spend more time communicating with the user, my supervisor and my assessor in order to get sufficient feedback. Those above are the experience I gained and how to avoid them in the future.

Appendix B – Project Schedule





Appendix C – Requirements Gathering

Project Propose Email

However, can I make a suggestion? What would be really useful is if there was a list of ale houses in and around Leeds with map links so we know where they are. If it is there and I missed it I apologise. Wouldn't be too difficult to put a database behind the site so one could do a postcode search for ale houses. Just a thought...

User Requirements

The problem owner of this project is Leeds CAMRA. Tony Jenkins is the project owner representative. Therefore, the user requirements for this project are from Tony Jenkins. As there is no formal interview has been hold between Tony and the developer, the following information is summary combination transcript of the meeting conversions and e-mail contacts with Tony Jenkins during the development process between 20/10/2006 and 10/.03/2007.

Q: What's the project about?

A: This is a web database application for Leeds pubs in a map. The system should use persistent data storage and have a graphical interface.

Q: What's this map system for?

A: The basic idea is to create a system which can display a list of ale houses in and around Leeds in a map. And, people can do a postcode search for ale house. And, the map should provide a zoom in and zoom out panel. It might be a good idea to display both map view and satellite view.

Q: What are the general requirements for this system?

A: This system should build with free techniques. It will be host on Linux, this server support PHP 4.3.10 – 16, and MySQL Ver. 12.22

Q: What functions will this system provide apart from search pubs from the map?

A: A system that will allow administrator to create/edit/delete pubs records in the map.

Q: I can produce function to let user register with this system and post comments on the page or maybe upload picture to the website, does Leeds CAMRA needs those kinds of functions?

A: No. We don't want people leave a message saying "This pub sacks".

Q: Apart from searchable map, login system for administrator and add/edit/delete functions for administrator, are there any other requirements?

A: I still think the best way to elicit requirements is to put together a quick example of what the system might do.

Q: Do you think the system is too simply. How about I use my creativity to create some functions, such as add to favourite and share with friends?

A: No, Leeds CAMRA doesn't want those functions. You can build a prototype of the system soon then show it to me as you claim it is easy and I give you more ideas.

Q: How many different levels of administration should I produce?

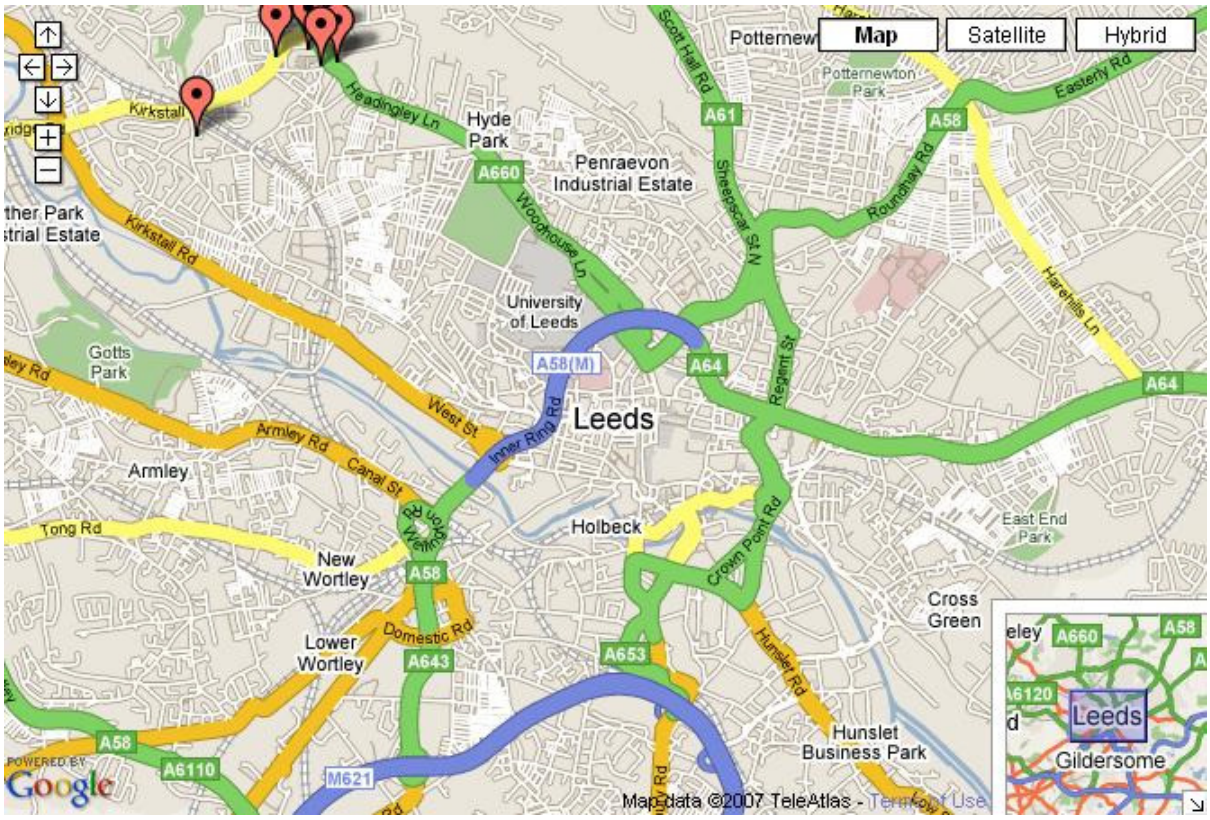
A: Maybe two, one normal administrator and the other is super administrator. The administrator can only maintain pubs records while the super administrator can maintain everything provided by the system.

Q: What information should I provide and where can I get the data?

A: Pubs' latitude, longitude, name, postcode, address, district, website, owned by, run by, phone, rank, picture, and note information should be provide by this system. The originally data can be collect from the Excel Spreadsheet and from the book - Leeds Beer Drinkers Companion.

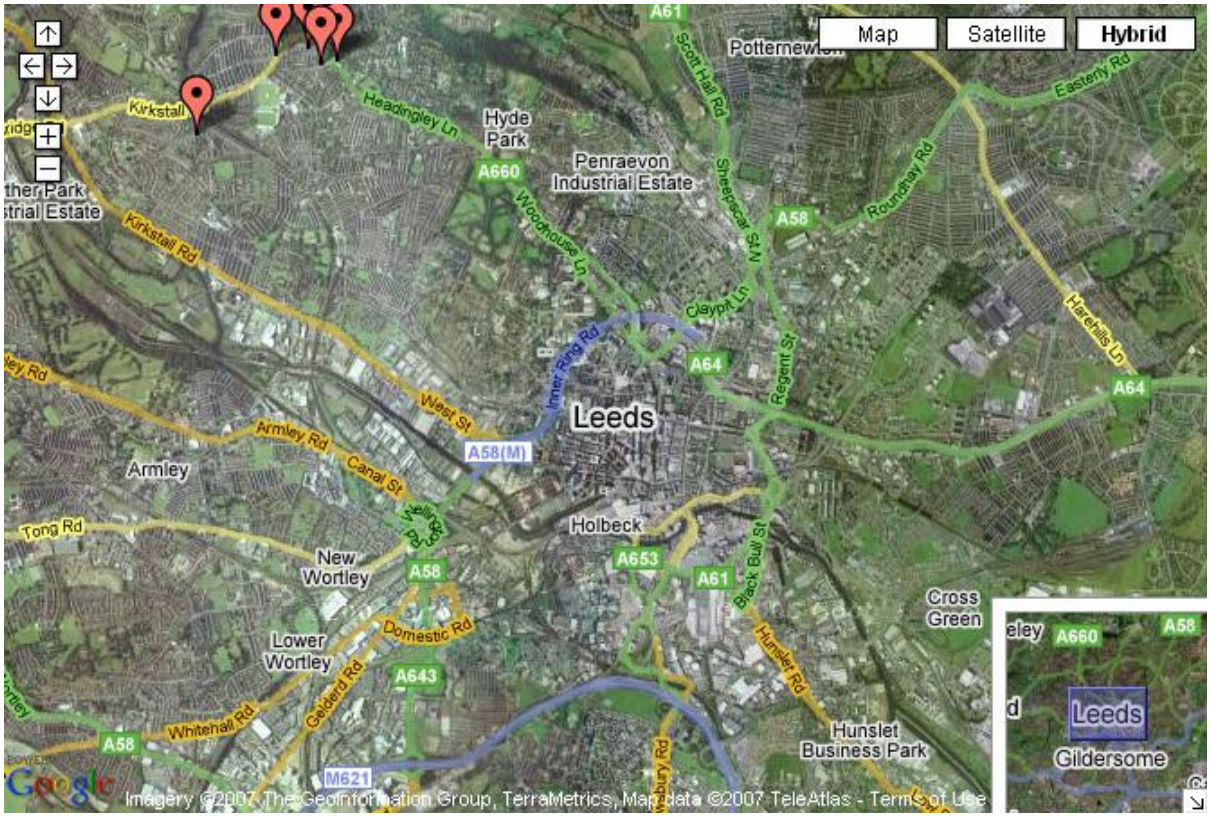
Appendix D – Three screens shoot of different views of the map

The map view



The satellite view

The hybird view



Appendix E – Initial SQL statements transcript

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

--
-- Create schema camra
--

CREATE DATABASE /*!32312 IF NOT EXISTS*/ camra;
USE camra;

--
-- Table structure for table `camra`.`default`
--

DROP TABLE IF EXISTS `default`;
CREATE TABLE `default` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `latitude` float default NULL,
  `longitude` float default NULL,
  `map_level` int(10) unsigned default NULL,
  `api` varchar(45) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `camra`.`default`
--

/*!40000 ALTER TABLE `default` DISABLE KEYS */;
/*!40000 ALTER TABLE `default` ENABLE KEYS */;



--
-- Table structure for table `camra`.`districts`
```

```
--

DROP TABLE IF EXISTS `districts`;
CREATE TABLE `districts` (
   `id` int(10) unsigned NOT NULL auto_increment,
   `district_name` varchar(45) default NULL,
   PRIMARY KEY    (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `camra`.`districts`
--

/*!40000 ALTER TABLE `districts` DISABLE KEYS */;
INSERT INTO `districts` (`id`,`district_name`) VALUES
  (1,'Aberford'),
  (2,'Allerton Bywater'),
  (3,'City Centre'),
  (4,'Headingley');
/*!40000 ALTER TABLE `districts` ENABLE KEYS */;

--
-- Table structure for table `camra`.`markers`
--

DROP TABLE IF EXISTS `markers`;
CREATE TABLE `markers` (
   `id` int(10) unsigned NOT NULL auto_increment,
   `latitude` float NOT NULL default '0',
   `longitude` float NOT NULL default '0',
   `name` varchar(45) NOT NULL,
   `postcode` varchar(45) default '',
   `website` varchar(45) default '',
   `owned_by` varchar(45) default '',
   `address` varchar(45) default '',
   `phone` varchar(45) default '',
   `rank_id` int(10) unsigned default '0',
   `picture` varchar(45) default NULL,
   `note` varchar(45) default NULL,
   `district_id` int(10) unsigned default NULL,
   `run_by` varchar(45) default NULL,
   PRIMARY KEY    (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `camra`.`markers`
--

/*!40000 ALTER TABLE `markers` DISABLE KEYS */;
INSERT INTO `markers` (`id`,`latitude`,`longitude`,`name`,`postcode`,`website`,`owned_by`,`address`,`phone`,`rank_id`,`picture`,`note`,`district_id`,`run_by`) VALUES
  (9,53.8208,-1.5815,'Arc','LS6 3JJ','','','19 Ash Road','01132752233',5,NULL,'',4,''),
  (10,53.8222,-1.5781,'Arcadia Ale & Wine Bar','LS6 2UE','','Market Town Taverns','Arndale Cenre, Otley Road','01132745599',5,NULL,'',4,'Market Town Taverns'),
  (11,53.8154,-1.5891,'Boston Exchange','LS4 2SE','','','44 St. Anns Lane','01132755404',4,NULL,'',4,''),
  (12,53.8208,-1.577,'Box','LS6 2AD','','','10 Otley Road','01132249266',5,NULL,'',4,''),
  (13,53.8203,-1.5785,'Headingley Taps','LS6 3HN','','','North Lane','01132200931',5,NULL,'',4,''),
  (14,53.8254,-1.5802,'New Inn','LS6 4DL','','','New Inn','01132249131',5,NULL,'',4,''),
  (15,53.8196,-1.5757,'Original Oak','LS6 2DG','','','2 Otley Road','01132751322',5,NULL,'',4,''),
  (16,53.8194,-1.5772,'Skyrack','LS6 3AW','','','2 St. Michaels Road','01132785836',5,NULL,'',4,''),
  (17,53.8279,-1.5828,'Three Horseshoes Hotel','LS16 5JG','','','98 Otley Road','01132757222',5,NULL,'',4,''),
  (18,53.8199,-1.5815,'Trio Bar & Grill','LS6 3HU','','','44 North Lane','01132036090',4,NULL,'',4,'');
INSERT INTO `markers` (`id`,`latitude`,`longitude`,`name`,`postcode`,`website`,`owned_by`,`address`,`phone`,`rank_id`,`picture`,`note`,`district_id`,`run_by`) VALUES
  (19,53.8279,-1.5828,'Woodies','LS16 5JG','','','104 Otley Road ','01132784393',5,NULL,'',4,'');
/*!40000 ALTER TABLE `markers` ENABLE KEYS */;

--
-- Table structure for table `camra`.`markers_rank`
--

DROP TABLE IF EXISTS `markers_rank`;
CREATE TABLE `markers_rank` (
   `id` int(10) unsigned NOT NULL auto_increment,
   `rank` varchar(45) NOT NULL,
   PRIMARY KEY    (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `camra`.`markers_rank`
--

/*!40000 ALTER TABLE `markers_rank` DISABLE KEYS */;
INSERT INTO `markers_rank` (`id`,`rank`) VALUES
```

```
  (1,'One Star'),
  (2,'Two Stars'),
  (3,'Three Stars'),
  (4,'Four Stars'),
  (5,'Five Stars');
/*!40000 ALTER TABLE `markers_rank` ENABLE KEYS */;

--
-- Table structure for table `camra`.`users`
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
   `id` int(10) unsigned NOT NULL auto_increment,
   `loginname` varchar(45) NOT NULL,
   `password` varchar(45) NOT NULL,
   `firstname` varchar(45) NOT NULL,
   `surname` varchar(45) NOT NULL,
   `level_id` varchar(45) NOT NULL default '',
   PRIMARY KEY    (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `camra`.`users`
--

/*!40000 ALTER TABLE `users` DISABLE KEYS */;
INSERT INTO `users` (`id`,`loginname`,`password`,`firstname`,`surname`,`level_id`) VALUES
  (1,'admin','admin','Jacky','Zhang','1'),
  (2,'test','test','Local','Admin','2');
/*!40000 ALTER TABLE `users` ENABLE KEYS */;

--
-- Table structure for table `camra`.`users_levels`
--

DROP TABLE IF EXISTS `users_levels`;
CREATE TABLE `users_levels` (
   `id` int(10) unsigned NOT NULL auto_increment,
   `level` varchar(45) NOT NULL,
   PRIMARY KEY    (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```
-- Dumping data for table `camra`.`users_levels`
--

/*!40000 ALTER TABLE `users_levels` DISABLE KEYS */;
INSERT INTO `users_levels` (`id`,`level`) VALUES
  (1,'Super Admin'),
  (2,'Admin');
/*!40000 ALTER TABLE `users_levels` ENABLE KEYS */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

Appendix F - Programming codes

JavaScript Validate the Digit

```
<Script language="javascript" type="text/javascript">
function checkForm()
{
   number = document.newMarker.phone.value;

   if(isNaN(number))
   {
      alert ("Please make sure the phone number is an integer");
      return false;
   }
   else return true;
}
</Script>
```

JavaScript Validate the Postcode

```
<Script language="javascript" type="text/javascript">
function checkPostcode()
{
   var filter = /[A-Z]{1,2}[0-9R][0-9A-Z]? [0-9][A-Z-CIKMOV {2}/;
   if (!filter.test(value))
      {return false;}
   return true;
}
</Script>
```

The example of connecting the MySQL database using PHP

```
<?php
  $host = "csdbdev.leeds.ac.uk";
  $username="efy2jz";
  $password="efy2jzefy2jz";
  $database="efy2jz_FYP";

  mysql_connect($host,$username,$password);
  mysql_select_db($database) or die( "Unable to select database");
  $query="SELECT * FROM markers";
  $result=mysql_query($query);

$num=mysql_numrows($result);

mysql_close();
```

```
 for ($i=0; $i<$num; $i++)
 {
  $id=mysql_result($result,$i,"id");
  $latitude=mysql_result($result,$i,"latitude");
  $longitude=mysql_result($result,$i,"longitude");
  $name=mysql_result($result,$i,"name");
  echo "addMarker($latitude, $longitude,'$name');";
 }
 ?>
```

Google maps application

```
<script src="http://maps.google.com/maps?file=api&amp;v=2&amp;key=<?php echo $api_key?>" type="text/javascript"></script>

<script type="text/javascript">
var centerLatitude = 53.80;
var centerLongitude = -1.55;
var startZoom = 13;
var map;

//function to create marker
function addMarker(latitude, longitude, description) {
var marker = new GMarker(new GLatLng(latitude, longitude));

GEvent.addListener(marker, 'click',
function() {
marker.openInfoWindowHtml(description);
}
);

map.addOverlay(marker);
}

//the main initial method
function init() {
if (GBrowserIsCompatible()) {
map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.addControl(new GOverviewMapControl());
map.setCenter(new GLatLng(centerLatitude, centerLongitude), startZoom);

//allow the user to click the map to create amarker.
```

```
GEvent.addListener(map, "click", function(overlay, latlng) {
//only perform the click if the window is closed and the click was directly on the map.
if(!overlay) {
//create an HTML DOM form element
var inputForm = document.createElement("form");
inputForm.setAttribute("action","");
inputForm.onsubmit = function() {storeMarker(); return false;};

//retrieve the longitude and lattitude of the click point
var lng = latlng.lng();
var lat = latlng.lat();

//We need DOM to create form element beacuse Gmarker.openInfoWindow() method expects
an HTML DOM as the second parameter
inputForm.innerHTML = '<fieldset style="width:150px;">'
+ '<legend>New Marker</legend>'
+ '<label for="name">Name</label>'
+ '<input type="text" id="name" style="width:100%;"/>'
+ '<input type="submit" value="Save"/>'
+ '<input type="hidden" id="longitude" value="' + lng + '"/>'
+ '<input type="hidden" id="latitude" value="' + lat + '"/>'
+ '</fieldset>';

map.openInfoWindow (latlng,inputForm);
}
});

function storeMarker(){
var lng = document.getElementById("longitude").value;
var lat = document.getElementById("latitude").value;

var getVars = "?name=" + document.getElementById("name").value + "&lng=" + lng +
"&lat=" + lat ;
}

 <?php
 $host = "csdbdev.leeds.ac.uk";
 $username="efy2jz";
 $password="efy2jzefy2jz";
 $database="efy2jz_FYP";

 mysql_connect($host,$username,$password);
 mysql_select_db($database) or die( "Unable to select database");
 $query="SELECT * FROM markers";
```

```
  $result=mysql_query($query);

$num=mysql_numrows($result);

mysql_close();

  for ($i=0; $i<$num; $i++)
  {
    $id=mysql_result($result,$i,"id");
    $latitude=mysql_result($result,$i,"latitude");
    $longitude=mysql_result($result,$i,"longitude");
    $name=mysql_result($result,$i,"name");
    echo "addMarker($latitude, $longitude,'$name');";
  }
  ?>
}
}

window.onload = init;
window.onunload = GUnload;
</script>
```

Search Pubs function in Public page

```
    if (strlen($name) >= 1 and strlen($postcode))
    {
      $query = "SELECT * FROM markers WHERE name like '%$name%' AND postcode
like '%$postcode%'";
    }
    elseif (strlen($name) >= 1)
    {
      $query = "SELECT * FROM markers WHERE name like '%$name%'";
    }
    elseif (strlen($postcode) >= 1)
    {
      $query = "SELECT * FROM markers WHERE postcode like '%$postcode%'";
    }
    else
    {
      $query = "SELECT * FROM markers";
    }
```

Appendix G- HTML W3C validation

This page is not Valid XHTML 1.0 Strict! (The index.php file)

The error message of the initial index.php page under XHTML 1.0 Strict standard validation is presented as follows.

Below are the results of checking this document for XML well-formedness and validity.

1. Warning *Line 39 column 75*: character "&" is the first character of a delimiter but occurred as data.

   ...Marker(53.8222, -1.5781,'Arcadia Ale **&** Wine Bar');addMarker(53.8154, -1.5891,

   This message may appear in several cases:

   - You tried to include the "<" character in your page: you should escape it as "<"
   - You used an unescaped ampersand "&": this may be valid in some contexts, but it is recommended to use "&", which is always safe.
   - Another possibility is that you forgot to close quotes in a previous tag.
2. Warning *Line 39 column 422*: character "&" is the first character of a delimiter but occurred as data.

   ...addMarker(53.8199, -1.5815,'Trio Bar **&** Grill');addMarker(53.8279, -1.5828,'Wo

   This message may appear in several cases:

   - You tried to include the "<" character in your page: you should escape it as "<"
   - You used an unescaped ampersand "&": this may be valid in some contexts, but it is recommended to use "&", which is always safe.
   - Another possibility is that you forgot to close quotes in a previous tag.
3. Error *Line 52 column 103*: there is no attribute "border".

   ...t; padding: 10px;" alt="logo" border="0" width="180px" height="50px">

You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the element-undefined error message for further information.

How to fix: check the spelling and case of the element and attribute, (Remember XHTML is all lower-case) and/or check that they are both allowed in the chosen document type, and/or use CSS instead of this attribute. If you received this error when using the <embed> element to incorporate flash media in a Web page, see the FAQ item on valid flash.

4. Error *Line 52 column 138*: end tag for "img" omitted, but OMITTAG NO was specified.

   ...r="0" width="180px" height="50px">

   You may have neglected to close an element, or perhaps you meant to "self-close" an element, that is, ending it with "/>" instead of ">".

5. Info *Line 52 column 23*: start tag was here.

   <a href="index.php"><img src="style/logo.jpg" style="float:left; padding: 10

6. Error *Line 71 column 14*: there is no attribute "name".

   <form name="search" action="results.php" method="post">

   You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the

"Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the element-undefined error message for further information.

How to fix: check the spelling and case of the element and attribute, (Remember XHTML is all lower-case) and/or check that they are both allowed in the chosen document type, and/or use CSS instead of this attribute. If you received this error when using the <embed> element to incorporate flash media in a Web page, see the FAQ item on valid flash.

7. Error *Line 72 column 32*: there is no attribute "height".

    <table width="295" height="81" border="0">

    You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

    This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the element-undefined error message for further information.

    How to fix: check the spelling and case of the element and attribute, (Remember XHTML is all lower-case) and/or check that they are both allowed in the chosen document type, and/or use CSS instead of this attribute. If you received this error when using the <embed> element to incorporate flash media in a Web page, see the FAQ item on valid flash.

8. Error *Line 74 column 20*: there is no attribute "width".

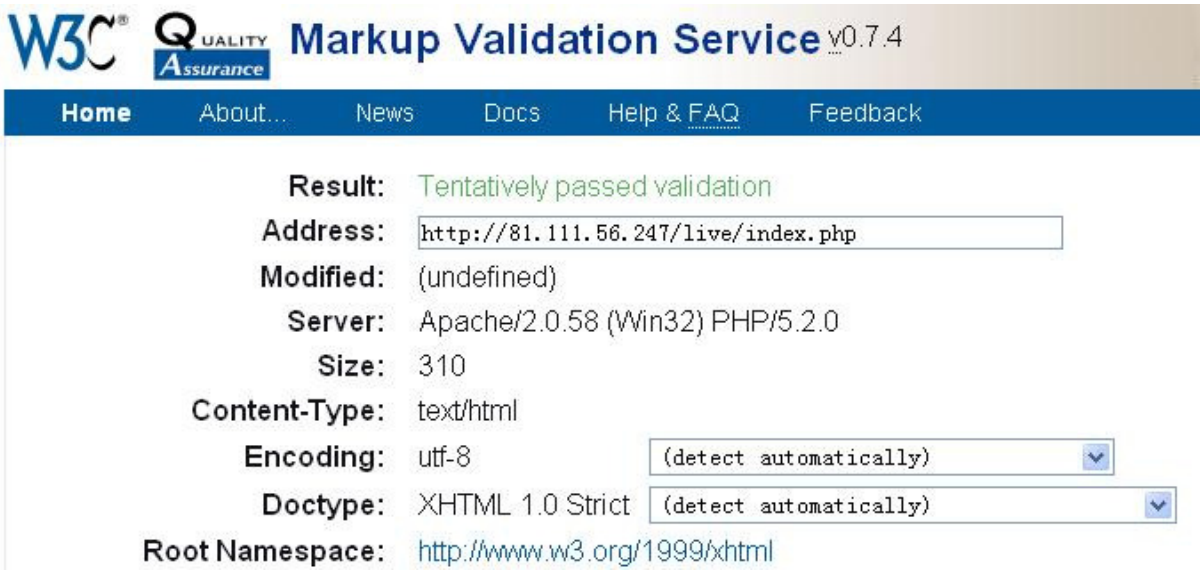
<td width="95">Name</td>

You have used the attribute named above in your document, but the document type you are using does not support that attribute for this element. This error is often caused by incorrect use of the "Strict" document type with a document that uses frames (e.g. you must use the "Transitional" document type to get the "target" attribute), or by using vendor proprietary extensions such as "marginheight" (this is usually fixed by using CSS to achieve the desired effect instead).

This error may also result if the element itself is not supported in the document type you are using, as an undefined element will have no supported attributes; in this case, see the element-undefined error message for further information.

How to fix: check the spelling and case of the element and attribute, (Remember XHTML is all lower-case) and/or check that they are both allowed in the chosen document type, and/or use CSS instead of this attribute. If you received this error when using the <embed> element to incorporate flash media in a Web page, see the FAQ item on valid flash.

This page is tentatively valid XHTML 1.0 Strict (The index.php file)


W3C QUALITY Assurance Markup Validation Service v0.7.4
Home About... News Docs Help & FAQ Feedback
Result: Tentatively passed validation
Address: http://81.111.56.247/live/index.php
Modified: (undefined)
Server: Apache/2.0.58 (Win32) PHP/5.2.0
Size: 310
Content-Type: text/html
Encoding: utf-8 (detect automatically)
Doctype: XHTML 1.0 Strict (detect automatically)
Root Namespace: http://www.w3.org/1999/xhtml

Appendix H - Test Plan

Unit Test				
No	**Task**	**Result**	**Problem**	**Corrective Action**
1	In the main page, put "ls6" in the postcode field and click search, all the pubs has "ls6" in the postcode should be displayed.	As expected		
2	In the main page, put "box" in the name field and click search, the pub name box should be displayed.	As expected		
3	In the main page, click search, all pubs should be displayed.	As expected		
4	Zoom in and zoom out the map	As expected		
5	Switch between different views of the map	As expected		
6	Move the small map	As expected		
7	Login the system using incorrect username: test and password: jacky, the user should not be able to login the system	As expected		
8	Login the system as a super administrator using correct username: admin and password: admin, the user should be able to login the system	As expected		
9	Login the system as normal administrator using correct username: test and password: test, the user should be able to login the system	As expected		
10	Click the Search/Edit Default Setting link in the secure page, user should be able to modify system default setting	Error	No time to implement	
11	Click the Create Default Setting link in the secure page, user should be able to create system default setting	Error	No time to implement	
12	Click Search/Edit admin user, user go into the search user page	As expected		
13	Click the search button in the search user page, all admin user will be displayed in the user results page	As expected		
14	Click the delete link in the user results page, it should pop up a warning JavaScript window	Error	Forgot to call the JavaScript method	Add a onsubmit event

15	Click the delete link in the user results page, the record should be successfully deleted	As expected		
16	Click the Create Admin User link, the user should be direct to a page asking new admin user's detail	As expected		
17	Click the submit button in the create admin user page, the new admin detail should be stored into the database	As expected		
18	Click the Search/Edit Marker link, the user should be direct to the search pubs page.	As expected		
19	Put woodies into the name search field in the pubs search page, and click the search button, the record of woodies should be presented	As expected		
20	Press the ID link in the pubs result page, the user should to direct to the edit pubs page with the selected pub's detail displayed	As expected		
21	Click the delete link in the pubs results page, it should pop up a warning JavaScript window	As expected		
22	Click the delete link in the pubs results page, the selected pub should be deleted	As expected		
23	Change the rank of the woodies pub from 5 to 4 and submit the change	As expected		
24	Change the address of woodies from 104 Otley Road to 105 Otley Road and submit the change	As expected		
25	Click the create marker link in the main secure page, the user should be direct to the create pubs page	As expected		
26	Click the search latitude/longitude button in the create pubs page, a window contains Multimap should be pop up.	As expected		
27	Put detail of the new pub into the create pubs create, such as Ad Lib, LS1 6DT and 01132426116	As expected		
28	Press the Secure navigation link which should direct the user back to the main secure page	As expected		
29	Press the Home navigation link which	As expected		

	should direct the user back to the main public page			
30	After press the home navigation link in the secure page, the user click the secure page which will link the user into the main secure page	As expected		
31	Press the Logout navigation link which should direct the user back to the main public page	As expected		
32	After press the logout navigation link in the secure page, the user click the secure page which will link the user into the login page	As expected		
33	Click the CAMRA logo, which can direct the user to the public main page	As expected		
34	Resize the window, the content should be adjusted accordingly as the system use the CSS layout	As expected		

User Acceptance Test	
Requirement	**Achieved?**
The system should be developed using free software and methods	Yes
The project should fit into the current host server which is Linux system with PHP 4.3.10-16 and MySQL Ver. 12.22	Yes
The system should provide a graphical user interface and the system's data should be stored in persistent database	Yes
The system should be able to be accessed from anywhere	Yes
Visitors will be able to search pubs in the active map by name and postcode	Yes
Administrators will be able to maintain records	Yes

Note: The test plan is a simply version as there is no time left to carry out a full in-depth test plan. However, this test plan has tested the most fundamental functions and usability of the system. Therefore, it can conclude that the system is success based on the above test.

Appendix I - Final Files List

File Name	File Detail
apikey.php	Store initial Google Maps API
detail.php	The detail of a particular pub
footer.php	The footer for public files
header.php	The header for secure files
index.php	The main system public page. It provides functions for the user to search pubs
login.php	Administrator login from this page
loginAct.php	The administrator login action page, this page checks whether the username and password provided by the user are correct
results.php	Display search results
secure/addMarker.php	Create the pubs record, the user provide pubs' detail here
secure/ addMarkerAct.php	Pubs' detail stores into database
secure/addSetting.php	Create the setting records, the user provide default latitude, default longitude, map level and API here
secure/addSettingAct.php	Default setting information stores into database
secure/addUser.php	Create the administrators records, the user provide administrators' detail here.
secure/addUserAct.php	Administrators' detail stores into database
secure/deleteMarker.php	Delete pubs
secure/deleteUser.php	Delete administrators
secure/editMarker.php	Edit pubs
secure/editMarkerAct.php	Edit pubs action page
secure/editSetting.php	Edit default setting (default latitude, default longitude, map level and API)
secure/editSettingAct.php	Edit default setting action page
secure/footer.php	The footer for secure page
secure/header.php	The footer for header page
secure/index.php	The main system secure page. It provides links to all secure functions.
secure/searchMarker.php	Search pubs
secure/searchMarkerResult.php	The results of pubs search
secure/searchUser.php	Search users
secure/searchUserResult.php	The results of users search
style/logo.jpg	The logo of the system
style/style.css	The cascading style sheet of the system
manual.doc	The user manual available to download

Appendix J - User Manual

Welcome to the Leeds CAMRA. This is a brief manual for the Web Information System which guide user how to use the system properly. This software is produced using a number of screen shoots of the system as the system is straight forward so there is no point writing lots of text to describe it.

Content

1. Introduction
2. Public Functions
   2.1 Search the pubs
   2.2 Login to the system secure page
3. Secure Functions
   3.1 System setting
      3.1.1 Search / Edit default setting
      3.1.2 Create default setting
   3.2 User management
      3.2.1 Search / Edit admin user
      3.2.1 Create admin user
   3.3 Marker management
      3.3.1 Search / Edit marker
      3.3.2 Create Marker
4. Conclusion

1. Introduction

Welcome to the manual of Web Information system for a Map of Leeds Pubs. The system of will provide information for a list of CAMRA specified ale houses in Leeds in a rich interface map. This system will allow users to search for specific bars with the map. A content management system for administrators is available for the user. This manual will explain all functions provided by the public pages and secure pages. Hope you have a detail understanding on how to use the system.

2. Public Functions

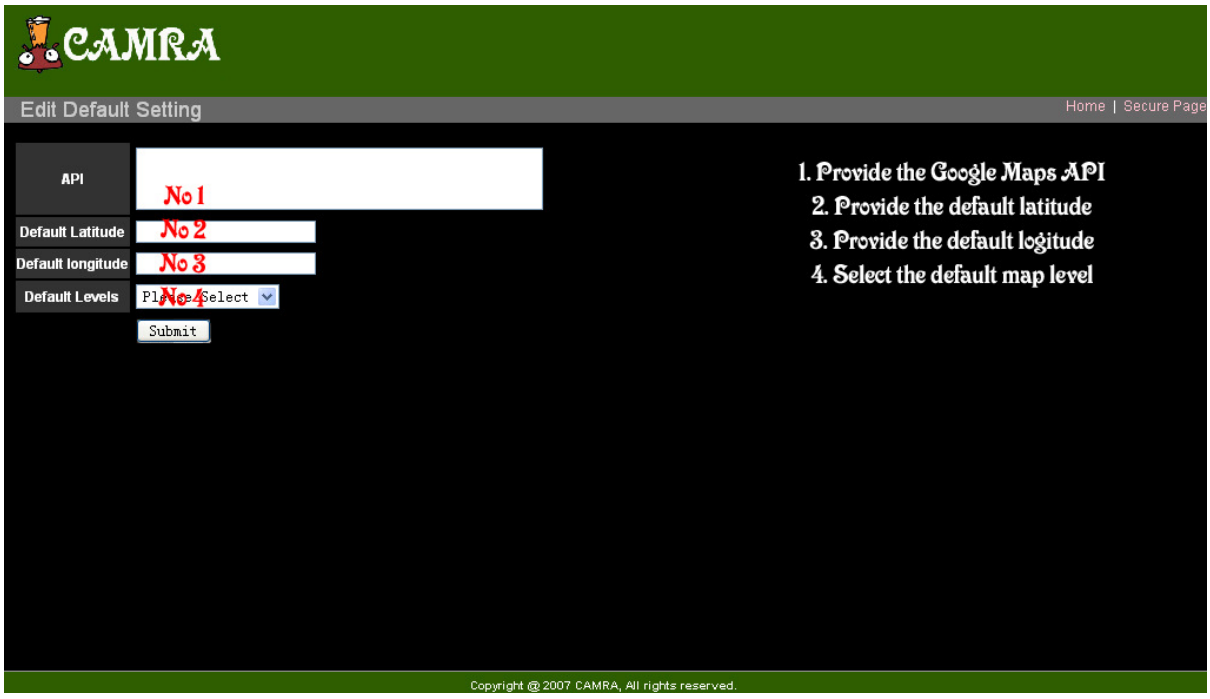
2.1 Search the pubs

2.2 Login to the system secure page

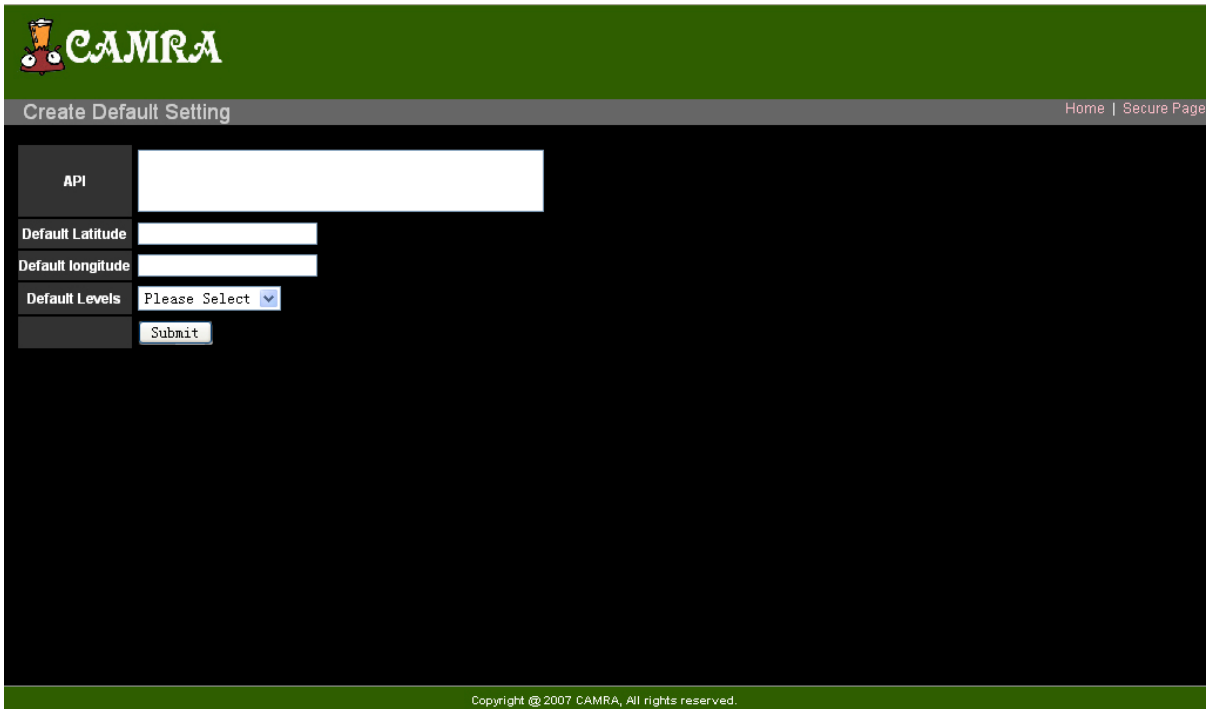
3. Secure Functions

3.1 System setting

3.1.1 Search / Edit default setting



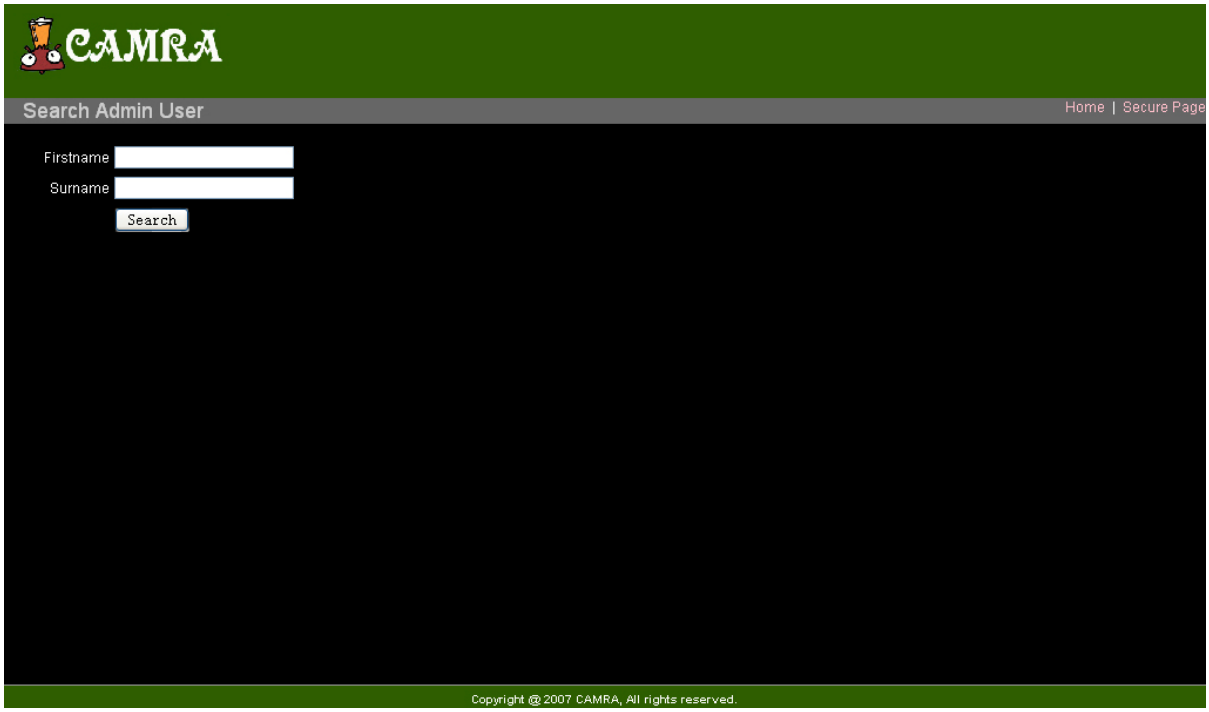
3.1.2 Create default setting



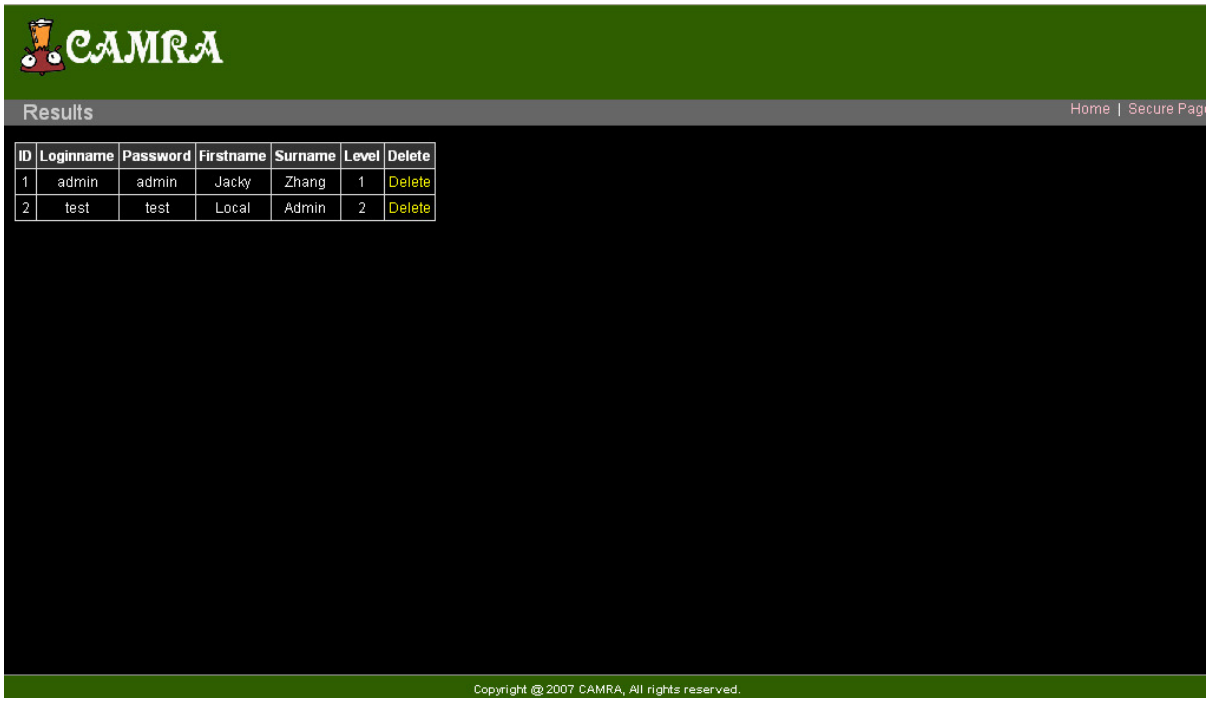
3.2 User management

3.2.1 Search / Edit admin user

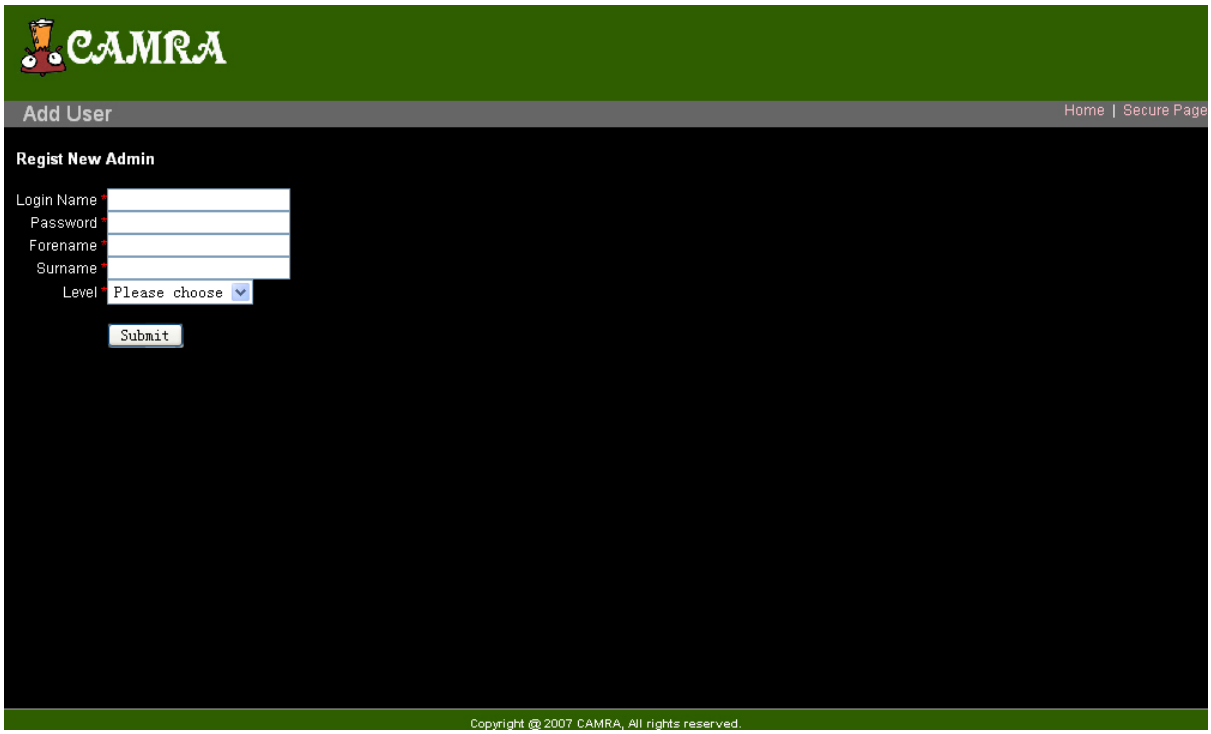
Search Page



Result Page


CAMRA
Results
Home | Secure Pag
ID Loginname Password Firstname Surname Level Delete
1 admin admin Jacky Zhang 1 Delete
2 test test Local Admin 2 Delete
Copyright @ 2007 CAMRA, All rights reserved.

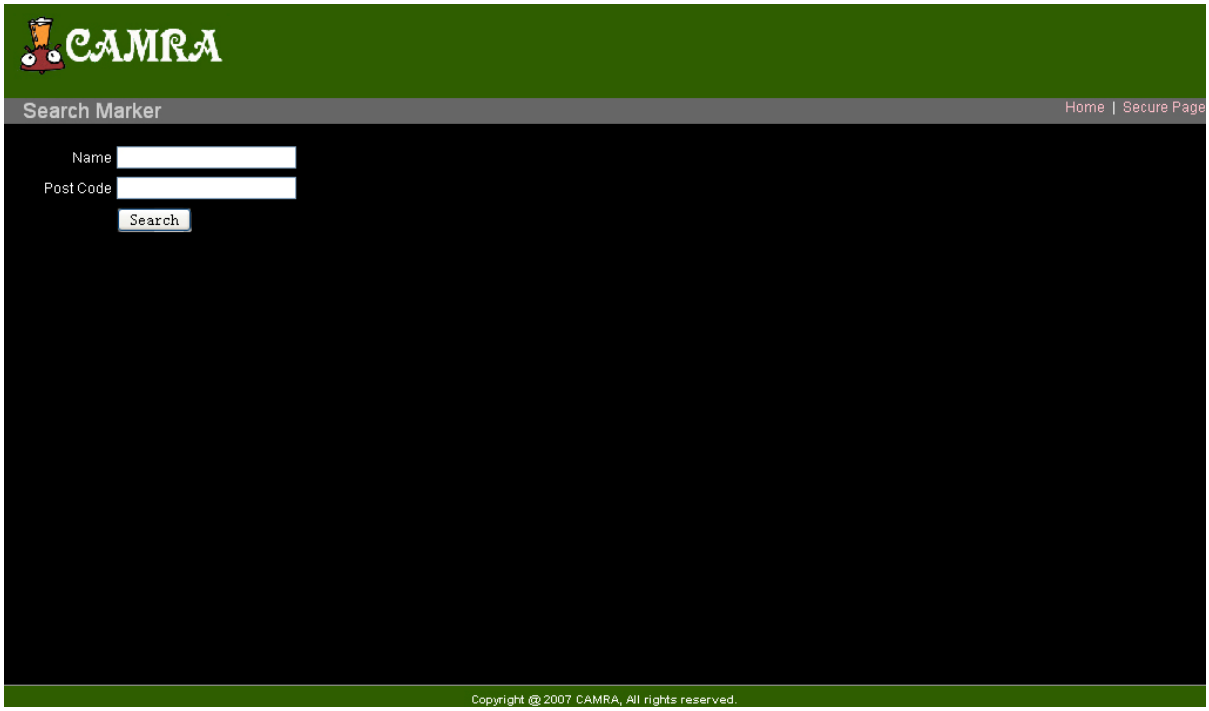
3.2.1 Create admin user



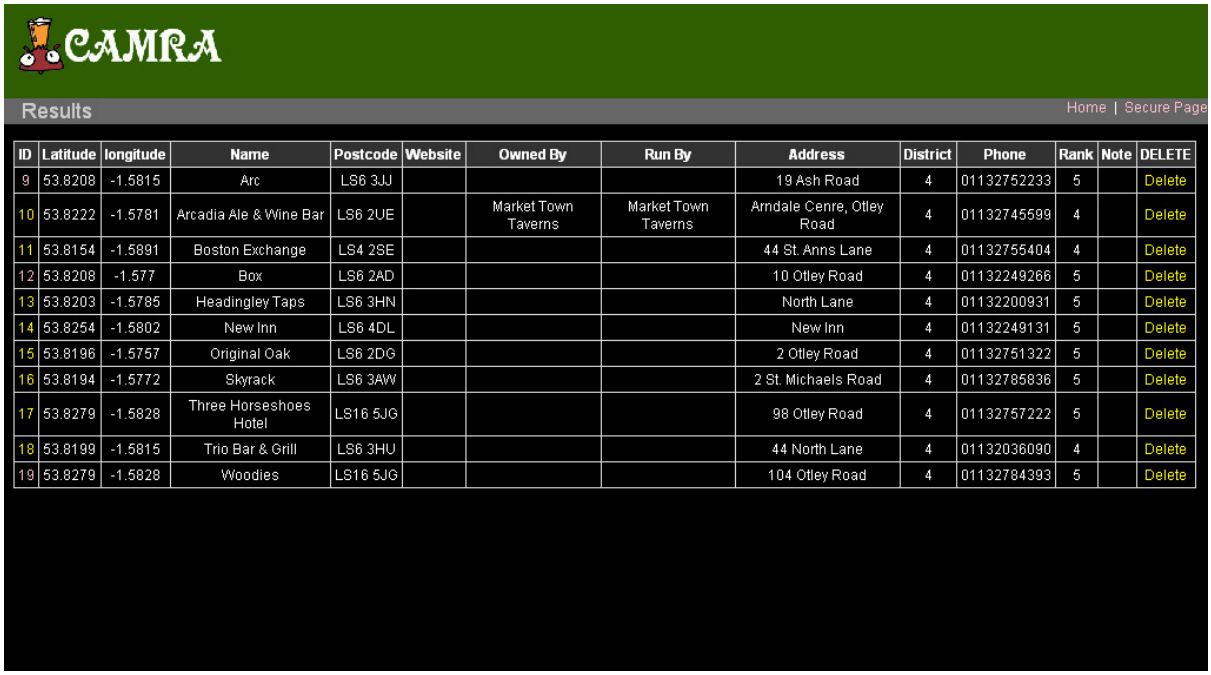
3.3 Marker management

3.3.1 Search / Edit marker

Search Page

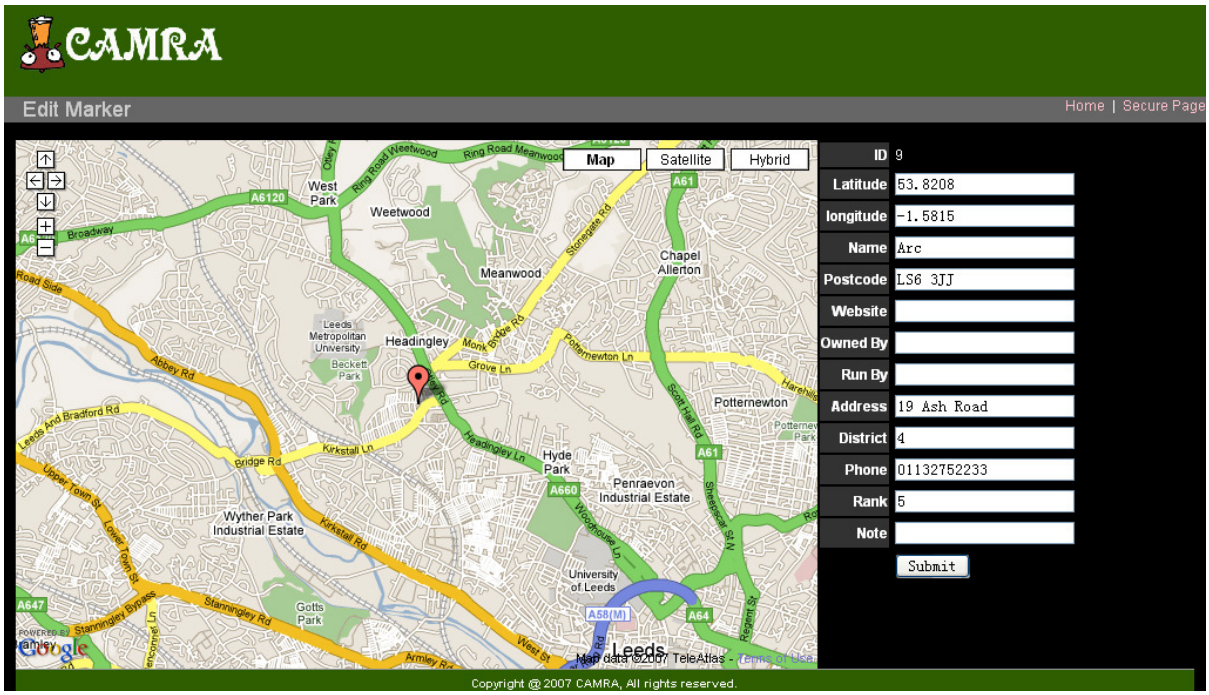


Result Page

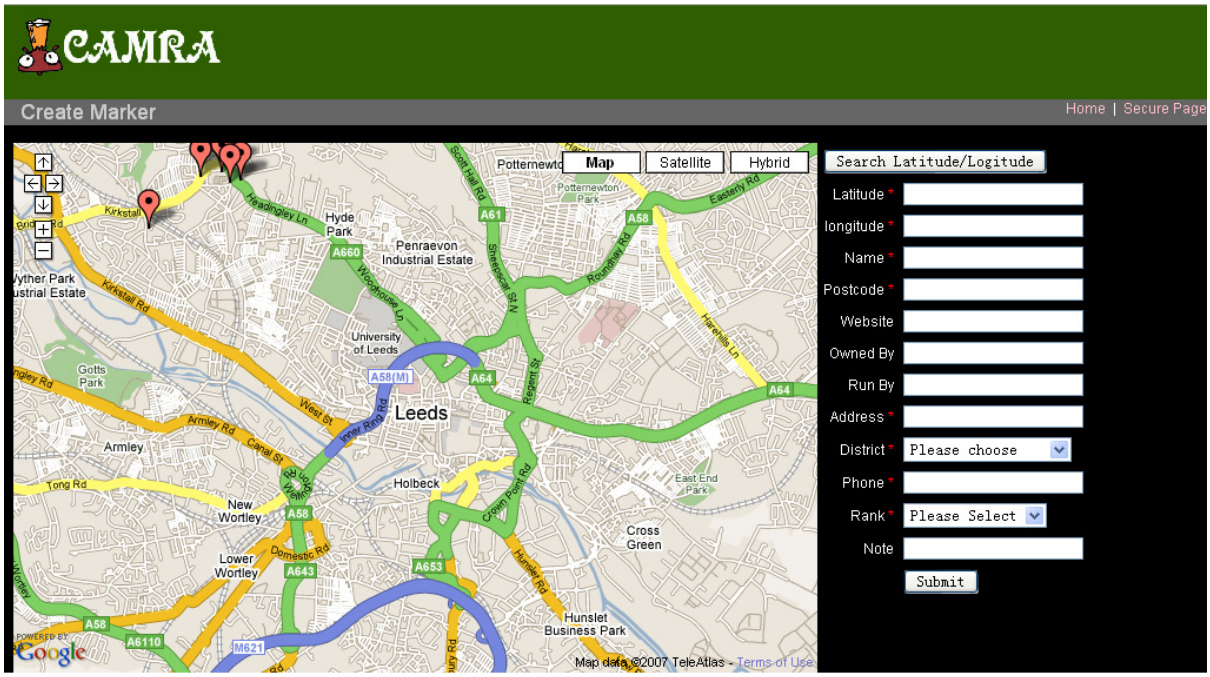


ID	Latitude	longitude	Name	Postcode	Website	Owned By	Run By	Address	District	Phone	Rank	Note	DELETE
9	53.8208	-1.5815	Arc	LS6 3JJ				19 Ash Road	4	01132752233	5		Delete
10	53.8222	-1.5781	Arcadia Ale & Wine Bar	LS6 2UE		Market Town Taverns	Market Town Taverns	Arndale Cenre, Otley Road	4	01132745599	4		Delete
11	53.8154	-1.5891	Boston Exchange	LS4 2SE				44 St. Anns Lane	4	01132755404	4		Delete
12	53.8208	-1.577	Box	LS6 2AD				10 Otley Road	4	01132249266	5		Delete
13	53.8203	-1.5785	Headingley Taps	LS6 3HN				North Lane	4	01132200931	5		Delete
14	53.8254	-1.5802	New Inn	LS6 4DL				New Inn	4	01132249131	5		Delete
15	53.8196	-1.5757	Original Oak	LS6 2DG				2 Otley Road	4	01132751322	5		Delete
16	53.8194	-1.5772	Skyrack	LS6 3AW				2 St. Michaels Road	4	01132785836	5		Delete
17	53.8279	-1.5828	Three Horseshoes Hotel	LS16 5JG				98 Otley Road	4	01132757222	5		Delete
18	53.8199	-1.5815	Trio Bar & Grill	LS6 3HU				44 North Lane	4	01132036090	4		Delete
19	53.8279	-1.5828	Woodies	LS16 5JG				104 Otley Road	4	01132784393	5		Delete

Edit Marker



3.3.2 Create Marker



4. Conclusion

The system should be developed using free software and methods and it is the current host server which is Linux system with PHP 4.3.10-16 and MySQL Ver. 12.22 (http://wwwdev.comp.leeds.ac.uk/efy2jz/CAMRA/). The guide on how to operate the system is demonstration above. Please contact the webmaster for further helps.

Hope you enjoy the system.

Leeds CAMRA